MNIV01

金沢大学計算機センター

# 利用の手引き

## ＴＳＳ実習編（初級用）

## ＴＳＳコマンド、データセット、実行のＡＢＣ

執筆者　　工学部　　助手

山崎　光悦

KANAZAWA UNIVERSITY

# はじめに

本利用の手引きは、既にセンタから発行されている利用の手引き〝ＴＳＳ実習編（初心者用）―ＦＯＲＴＲＡＮユーザのためのＴＳＳ入門―〟（中島　恵美著）〔１〕（注１）あるいは〝ＴＳＳ端末によるバッチ処理―フルスクリーン機能と出力検索―〟（関崎　正夫著）〔２〕を片手に、ディスプレイ端末に向ってＴＳＳを数回以上経験した利用者（ユーザ）を対象としている。したがって利用者は、フルスクリーン機能によるプログラム編集（ＥＤＩＴ）とサブミットジョブの依頼あるいはＲＵＮサブコマンド（言葉の定義は本文中で記述する）による実行などに関するある程度の概念を修得済みであるとして記述されている。さらに利用の手引き　〝バッチからＴＳＳへ〟（山崎　光悦著）〔３〕に記述されている程度のＴＳＳの概念およびデータセットに関する知識も必要である。

第１章のＴＳＳコマンド、第２章のデータセットに関する記述は、利用者が富士通発行のＴＳＳ関係のマニュアルを読む際に役に立つと思われる基礎的事項を集約してある。記述が単調なため〝忍耐〟を必要とするであろう。第３章では主にデータセット管理に関するコマンド、第４章では端末に向って即時実行するための種々のコマンドの使用方法を中心に例をまじえて解説する。したがって、第３章、第４章は一読した後、端末に向ってキーボードをたたいた方が修得し易いであろう。

最初に述べた二つの手引きと本手引きの内容を習得すれば、ＴＳＳに関する〝基本的な知識〟はほとんど網羅されたことになろう。さらに詳しい知識を必要とするユーザのために各部の記述の末尾に関連の利用の手引きおよびマニュアルを掲げるよう努めた。

（注１）　〔　〕内の数字は末尾の一覧に掲げた計算機センター発行の利用の手引き又は富士通発行のマニュアルの番号を表わす。

# 第1章　　ＴＳＳコマンド

ＴＳＳでは、利用者（ユーザ）が端末を使って計算機と会話形式で作業を進めるためにＴＳＳコマンドと呼ばれる計算機に与える命令言語が用意されている。このコマンドによって端末から計算機に作業依頼や問い合せをすると、計算機は必要な処理を行ってその結果やメッセージを出力する。利用者はその出力結果を見て、再びコマンドやデータを入力しながら順次作業を進めることができる。

## 1－1　　コマンドの種類

ユーザが用いるＴＳＳコマンドは、機能に応じて１０種類に分類される。それを表１に示す。またＥＤＩＴ、ＴＥＳＴＦＯＲＴ、ＯＵＴＰＵＴ、ＳＯＲＰなどのコマンドでは、更に詳しい機能を付加するためにそのコマンド下でのみ有効なコマンドが用意されている。それをサブコマンドと呼ぶ。ユーザは、与えようとする命令がコマンドかサブコマンドかをはっきりと区別し、また端末がどのような状態でそれを入力できるのかを常に意識する必要がある。

ＥＤＩＴコマンドの下で利用できるサブコマンドとして表２に示すものが準備されている。これらのコマンド、サブコマンドの利用法に関するやさしい解説は前述の利用の手引き〔１〕、〔２〕を参照されたい。更に詳しい使い方を習得されたいユーザは富士通発行のマニュアル〝ＴＳＳコマンド文法書〟〔４〕を参照されたい。なお、ディスプレイ型端末に有効なＥＤＩＴのフルスクリーン用サブコマンド（ＦＳサブコマンド）の下でのみ有効な各種の行編集サブコマンド群があるが、これを表３に示す。使い方の詳細は利用の手引き〔２〕及び〝ＴＳＳテキスト編集使用手引書　ＦＳＯ編〟〔５〕を参照されたい。表１～３に掲げたコマンド、サブコマンドはユーザが既に利用したことのあるものも多いはずだが、それらがコマンドなのかあるいはサブコマンドなのかをここで改めて確認されたい。

一方、出力検索及びその印刷に関するＳＯＲＰコマンド及びそのサブコマンド、ＤＳＰＲＩＮＴコマンドに関する解説は利用の手引き〔２〕を参照されたい。詳細は〝ＴＳＳ　ＳＯＲＰコマンド使用手引書〟〔６〕、〝ＴＳＳデータセットプリント（ＤＳＰＲＩＮＴ）使用手引書〟〔７〕をご覧頂きたい。またディバッグエイド用コマンド　（ＴＥＳＴＦＯＲＴ）に関する解説は利用の手引き〝ＴＳＳ実習編（中級用）―インタラクティブディバック―〟（関崎　正夫著）〔８〕を、また文法の詳細は〝ＦＯＲＴＲＡＮ７７　インタラクティブディバック使用手引書〟〔９〕を参照されたい。

表1　ＴＳＳコマンド一覧

| 種　別 | コマンド名 | 機　能　概　略 |
|---|---|---|
| （1）セション制御用 | LOGON | 新しいセション（注1）を開設し、各種コマンドの入力を可能とする。 |
| | LOGOFF | セションを終了する。 |
| （2）テキスト編集用 | EDIT | 各種ソースプログラム、データ、コマンドやサブコマンド、ジョブ制御文などのテキストからなるデータセットの作成、編集を行う。 |
| | MERGE | データセットの全体又は一部の内容を、別のデータセット中に組み込む。 |
| （3）プログラム処理用 | RUN | ソースプログラムを即時翻訳、実行する。 |
| | FORT77 | ＦＯＲＴＲＡＮ７７コンパイラによりフォートランソースプログラムを翻訳し、オブジェクトモジュール（注2）を作成、保存する。 |
| | LINK | リンケージエディタにより、オブジェクトモジュールを結合、編集してロードモジュール（注2）を作成、保存する。 |
| | CALL | ロードモジュールを実行する。 |
| | LOADGO | ローダにより、オブジェクトモジュールを結合編集して実行する。 |
| | FORT7CLG　＊ | ＦＯＲＴＲＡＮ７７コンパイラによる翻訳、結合編集、実行を可能とする。ロードモジュール（相対形式あるいは実行形式）の保存、実行も容易である。 |
| | CONVERT | フォートランソースプログラムの自由形式／標準形式の変換を行う。 |
| （4）データセット管理用 | ALLOCATE | データセットを割り当てる。 |
| | FREE | 割り当てられたデータセットを解放する。 |
| | ATTRIB | データセットの属性を登録する。 |
| | CONDENSE | 区分データセットを圧縮し、未使用領域を利用可能とする。 |
| | COMP　＊ | 区分データセットを圧縮し、未使用領域を解放する（ＣＯＮＤＥＮＳＥコマンドより便利）。 |

表1　ＴＳＳコマンド一覧

| | | | |
|---|---|---|---|
| | DELETE | | データセットまたは区分データセットのメンバを消去する |
| | RENAME | | データセット名または区分データセットのメンバ名を変更する |
| | COPY | | データセット又は区分データセットのメンバを複写する。 |
| | DSPRINT | | データセットの内容を編集してプリンタに出力する。 |
| | LIST | | データセットの内容を端末に出力する。 |
| | LISTALC | | 割り当てられたデータセットの一覧を出力する。 |
| | LISTV | ＊ | データセットの属性、占有トラック量などを出力する。 |
| | UCFL | ＊ | ユーザが占有する各データセットのトラック量、課金情報など各種利用者情報を出力する。 |
| | LISTCAT | | カタログに登録されたデータセットの一覧を出力する。 |
| | LISTDS | | データセットの属性、区分データセットのメンバ名を出力する。 |
| | PROTECT | | データセットをパスワードで保護する。 |
| (5) ディバッグエイド用 | TESTFORT | | ＴＥＳＴオプション付きで翻訳されたフォートランプログラムのオブジェクトモジュールあるいはロードモジュールを実行する際に、会話的なディバッグの手段を与える。 |
| (6) コマンドプロシジャ用 | EXEC | | コマンドプロシジャを実行する。 |
| | コマンドプロシジャ文（コマンド文 | | 詳細は〝ＴＳＳコマンド文法書〟〔4〕を参照されたい。 |
| | 、ＥＮＤ文、ＰＲＯＣ文、…） | | |
| (7) 会話型リモートバッチ | SUBMIT | | バッチジョブ（サブミットジョブ）の処理を依頼する。 |
| | STATUS | | 依頼したバッチジョブの処理状況を問い合わせ |

表1　ＴＳＳコマンド一覧

| | | | |
|---|---|---|---|
| 用 | | | る。 |
| | SORP | | バッチジョブの出力結果を検索する。 |
| | CANCEL | | 依頼したバッチジョブを取り消す。 |
| | OUTPUT | | バッチジョブの処理結果を端末又はデータセットに出力する。 |
| （8）端末制御用 | TERMINAL | | アテンション（注３）その他の端末特性を変更する。 |
| | PROFILE | | 文字消去キー、行消去キーの定義など、ユーザ端末の操作特性を定義する。 |
| | CANPR | ＊ | ＤＳＰＲＩＮＴコマンドによってプリンタ出力中のものを中止する。 |
| （9）システム状態表示用 | DAJ | ＊ | 実行中のバッチジョブのクラス別の情報を得る。 |
| | DQ | ＊ | バッチジョブとＴＳＳの稼動状態を表示する。 |
| | DAT | ＊ | ＴＳＳの使用者リストを出力する。 |
| | DAI | ＊ | 使用可能なジョブクラスを表示する。 |
| （１０）その他 | HELP | | コマンドの一覧やその構文、機能、オペランドの説明を出力する。 |
| | TIME | | ログオン時からのＣＰＵ時間やイラップス時間を得る。 |
| | CTSU | ＊ | 端末が作動不能になった場合、他の端末からそのＴＳＳセションを強制終了する。 |

＊　：　コマンド名の後に*印の付されているものはセンタで開発されたコマンドであり、詳細はセンタ発行の利用の手引き又は速報を参照されたい。〝ＴＳＳコマンド文法書〟〔４〕には記載されていない。

（注１）セション：ログオンからログオフまでの端末ユーザがサービスを受けている期間。

（注２）オブジェクトモジュール、ロードモジュール：コンパイラによって翻訳されたアセンブラ形式のプログラムをオブジェクトモジュール、さらにそれをリンケージエディタによって編集したものをロードモジュールという。

（注３）割り込み（アテンション）：コマンド、プログラム処理の実行を中断させること。

表2　ＥＤＩＴコマンドの主なサブコマンド一覧（ＦＳ機能によって代用されるものは除く）

| サブコマンド名 | 機　能　概　略 |
|---|---|
| ALLOCATE | データセットを割り当てる。 |
| FREE | 割り当てたデータセットを解放する。 |
| ATTRIB | データセットの属性を登録する。 |
| INPUT | 連続した行の入力を可能とする。 |
| SCAN | ソースプログラムの構文チェックを行う。 |
| MERGE | 他のデータセットから編集用データセットへ一連の行を組み込む。 |
| BOTTOM | 行指標を最下行へ移す。 |
| TOP | 行指標を先頭の行へ移す。 |
| FIND | 指定文字列を持つ行へ行指標を移す。 |
| FS | 画面編集を可能とする（フルスクリーン機能付きのディスプレイ端末にのみ有効）。 |
| RUN | 編集中のソースプログラムを翻訳、実行する。 |
| SUBMIT | バッチジョブの処理を依頼する。 |
| EXEC | コマンドプロシジャを実行する。 |
| RENUM | 行番号を付け直す。 |
| UNNUM | 行番号を削除する。 |
| HELP | ＥＤＩＴのサブコマンド一覧やその構文、機能、オペランドの説明を出力する。 |
| PROFILE | ユーザの端末属性の一部を変更する。 |
| TABSET | 論理タブの設定または変更を行う。 |
| AUTOSAVE | 編集用データセットの内容を自動的に保存する。 |
| SAVE | 編集中のデータセットを保存する。 |
| END | ＥＤＩＴコマンドの編集処理を終了する。 |

(注)　上記の表のサブコマンドの他にプリンタ端末に有効な各種サブコマンドがある。またＡＬＬＯＣＡＴＥ、ＡＴＴＲＩＢ、ＦＲＥＥ、ＭＥＲＧＥ、ＰＲＯＦＩＬＥ、ＳＵＢＭＩＴの各サブコマンドは、コマンドと同様の機能を有する。

表3．1　ＦＳ（フルスクリーン）サブコマンド下で有効なサブコマンドの一覧
ＦＳ画面の行番号フィールドの左端に入力する（行サブコマンド）

| サブコマンド名 | 名　称 | 機　　能 |
|---|---|---|
| I　n | insert | 新たな行をサブコマンド入力行の後にｎ行挿入する。 |
| D<br>DD | delete | Ｄを入力した行あるいはＤＤ～ＤＤで囲んだ複数行を消去する。 |
| R　n<br>RR　n | repeat | Ｒを入力した行あるいはＲＲ～ＲＲで囲んだ複数行をすぐ後にｎ回挿入する。 |
| C<br>CC | copy | Ｃを入力した行あるいはＣＣ～ＣＣで囲んだ複数行の複写を、Ａを入力した行の次あるいはＢを入力した行の前に作成する。 |
| M<br>MM | move | Ｍを入力した行あるいはＭＭ～ＭＭで囲んだ複数行を、Ａを入力した行の次あるいはＢを入力した行の前へ移動する。 |
| A | after | Ｃ、Ｍで指定した行をこの行の後に挿入する。 |
| B | before | Ｃ、Ｍで指定した行をこの行の前に挿入する。 |
| <　n<br><<　n | left | ＜を入力した行あるいは＜＜～＜＜で囲んだ複数行の内容をｎカラム左にシフトとする。 |
| >　n<br>>>　n | right | ＞を入力した行あるいは＞＞～＞＞で囲んだ複数行の内容をｎカラム右にシフトする。 |
| X<br>XX | exclude | Ｘを入力した行あるいはＸＸ～ＸＸで囲んだ複数行を表示させないようにする。 |
| S | show | Ｘ、ＸＸ～ＸＸで表示させないようにした行を再表示させる。 |

（注）　ｎは行サブコマンドの繰り返し数

表3．2　ＦＳ（フルスクリーン）サブコマンド下で有効なサブコマンドの一覧
ＦＳ画面のサブコマンド入力行（第二行目）に入力する

| コマンド名 | 機　能 |
|---|---|
| LOCATE　m | ＦＳ画面をｍで指定した行番号が先頭になるように移動する。 |
| RESET | 実行が保留になっている行サブコマンドを無効にする。 |

## 1－2　コマンドとモード

ＬＯＧＯＮコマンドによってセションを開設後、端末は各種のコマンドが入力可能な状態となる。この状態をコマンドモードと呼ぶ。コマンドモードでは入力した各種のコマンド（ＥＤＩＴ、ＴＥＳＴＦＯＲＴ、ＳＯＲＰコマンドなどサブコマンドを有するものを除く）を実行した後、必ず端末に〝ＲＥＡＤＹ〟というメッセージが表示される。この〝ＲＥＡＤＹ〟の表示によってコマンドモードであることが確認でき、それゆえコマンドモードをレディモードとも呼ぶ。

また、サブコマンドをもつコマンドを入力した場合は、その後端末はそのコマンドの下で有効な各種サブコマンドを入力できる状態となる。この状態をサブコマンドモードと呼ぶ。端末がサブコマンドモードであることは、一般にそのコマンド名が端末に出力されることによりわかる。例えば、ＥＤＩＴのサブコマンドモードでは〝ＥＤＩＴ〟（ＥＤＩＴコマンドを簡略形〝Ｅ〟と入力した場合は〝Ｅ〟と表示）というメッセージが出力される。

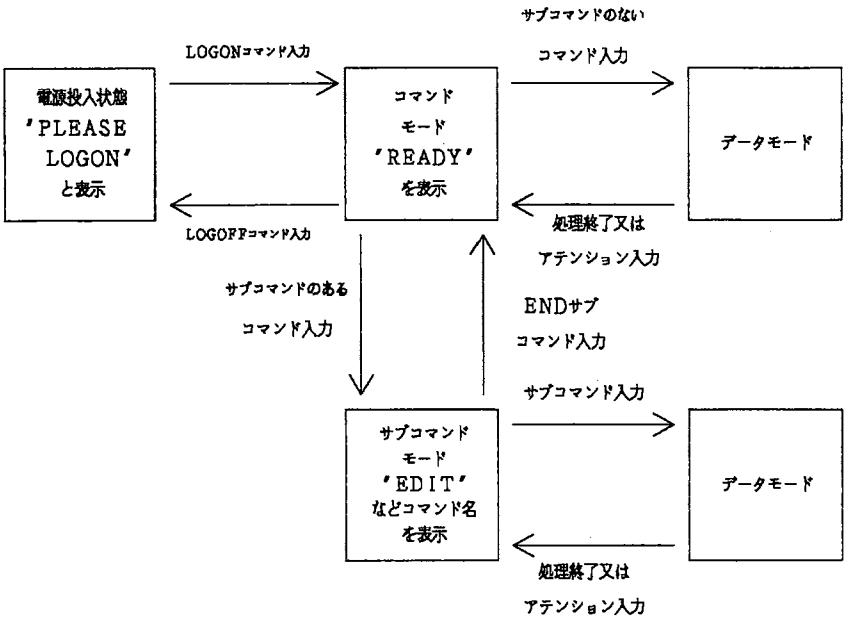



図1　コマンドモード、サブコマンドモード、データモードの関係

さらにコマンドあるいはサブコマンドを入力した後、端末が必要なデータなどの入力待ち状態となる場合がある。この状態をデータモードと呼ぶ。端末がデータモードの状態にあることは、通常行番号などが出力されることによってわかる。一方、コマンドやサブコマンドを入力した後、コマンド名に続いて入力したパラメータ（オペランド）に誤りや不正があるとその旨のメッセージが出力され、

端末ユーザに対して正しいパラメータの再入力を促す。この状態もデータモードの一種である。

以上述べたように端末はコマンドモード、サブコマンドモード、データモードの三つのモードの移り変りをくり返しながら、作業が進められる。これらの移り変りの関係およびモード変更に関与するコマンド、サブコマンドの対応関係を図１に示す。さらに実際の使用例での端末のモードの変遷を図２に示す。同図中の下線部分はユーザが端末のキーボードから入力したことを示す（以後同様とする）。また＜＋Ｅ＋＞記号は入力キー（ＥＮＴＥＲキー又はＲＥＴＵＲＮキー）の押下を意味する。

以上の記述からも明らかなようにユーザは端末がどのモードにあるかを認識し、コマンド、サブコマンドあるいはデータのいずれを入力すればよいかを判断しなければならない。

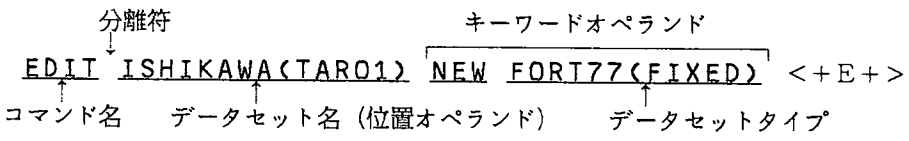
## １－３　コマンド、サブコマンドの入力形式

ＴＳＳコマンドを端末から入力するとき、コマンド名（あるいはサブコマンド名）とそのコマンドの機能の処理を実行する上で必要となる具体的なパラメータを指示するオペランドを入力する。コマンド名は全てのオペランドに先行して記述する必要がある。またオペランドには、指定する順序があらかじめ定まっている位置オペランドと、指定順序がなく特定の文字列を入力することによって認識されるキーワードオペランドとがある。位置オペランドは、データセット名や行番号の指定のように端末からユーザが任意の文字列を入力する必要があり、省略できないものが多い。一方、キーワードオペランドには互いに排他的なものとして定義されており、指定の際に選択する必要のあるものが多い。この場合、指定を省略すると標準値が自動的に選択されるものが多い。更に、キーワードオペランドには値を持つものもあり、その場合キーワードの直後に括弧でくくって指定する。

また、コマンド名とオペランド、オペランドとオペランドの区切りには１つ以上の分離符を用いる。分離符としてはブランクまたはコンマ″，″を用いることができる。

以上のことをＥＤＩＴコマンドの例でみると次のようになる。

（例）




## １－４　データセット名の指定

ＴＳＳで取り扱うデータセット名は、命名規約に従って次のように付けられる。

ユーザ識別修飾子．ユーザ指定名．内容識別修飾子

（例）　AB9999.KANAZAWA.FORT77

```
#    JCB 931I PLEASE LOGON
#   LOGON TSS AB9999 <+E+>
#    ***  10 GATSU NO TEIKIHOSHI WA 6 KA DESU          ****
#    ****                ................              ****
#    ENTER CURRENT PASSWORD FOR AB9999-
¥   ﾊﾟｽ ﾜｰﾄﾞ<+E+>
#    AB9999 LOGON IN PROGRESS AT 09:37:12 ON AUGUST 2, 1981
#    :
#    :
#    READY
#   EDIT KANAZAWA(TARO) NEW FORT77(FIXED) <+E+>
¥    INPUT
¥    00010  110 READ(5,*,END=120) A,B <+E+>
¥    00020      WA=A+B <+E+>
¥    00030      SA=A-B <+E+>
¥    00040      SEKI=A*B <+E+>
¥    00050      SYO=A/B <+E+>
¥    00060      WRITE(6,*) A,B,WA,SA,SEKI,SYO <+E+>
¥    00070      GO TO 110 <+E+>
¥    00080  120 STOP <+E+>
¥    00090      END <+E+>
¥    00100 <+E+>
%    EDIT
%   RUN <+E+>
%    FORTRAN77 COMPILER ENTERED
%    END OF COMPILATION
¥    00010 ?
¥   1.,2. <+E+>
¥      1.00000000 2.00000000 3.00000000 -1.00000000 2.00000000 0.50000000
¥    00010 ?
¥   5.6,8. <+E+>
¥      5.60000038 8.00000000 13.6000004 -2.39999962 44.8000031 0.70000048
¥    00010 ?
¥   /* <+E+>
¥    END OF GO, SEVERITY CODE=00
%    EDIT
%   SAVE <+E+>
%    SAVED IN DATA SET 'AB9999.KANAZAWA.FORT77(TARO)'
%    EDIT
%   END <+E+>
#    READY
#   LOGOFF <+E+>

      # : ｺﾏﾝﾄﾞﾓｰﾄﾞ   ¥ : ﾃﾞｰﾀﾓｰﾄﾞ   % : ｻﾌﾞｺﾏﾝﾄﾞﾓｰﾄﾞ
```

図2　モードの移り変り

これを完全修飾名という。ユーザ識別修飾子とはセンタから利用者に与えられた課題番号である。ユーザ指定名は一般名とも言い、英文字で始まる８文字以内の任意の英数文字列を利用者が指定する。また、内容識別修飾子とはデータセットの内容の種類を示すためにシステムが定めた特定の文字列か

らなり、データセットタイプとも呼ぶ（厳密には内容識別修飾子とデータセットタイプは完全に一致しないが、その対応関係を表４に示す）。

表４　主な内容識別修飾子の一覧

| データセットタイプ | 内容識別修飾子 | データセットの内容 |
|---|---|---|
| FORT77(FREE) | FORT77 | 自由形式のＦＯＲＴＲＡＮ７７のソースプログラム |
| FORT77(FIXED) | FORT77 | 標準形式のＦＯＲＴＲＡＮ７７のソースプログラム |
| CNTL | CNTL | ＦＩＢ（サブミット）用ジョブストリーム |
| DATA | DATA | 大文字のデータテキスト |
| OBJ | OBJ | オブジェクトモジュール |
| LOAD | LOAD | ロードモジュール |
| OUTLIST | OUTLIST | ＳＹＳＯＵＴデータ |
| CLIST | CLIST | コマンドプロシジャ |

ＴＳＳコマンドあるいはサブコマンドの位置オペランドとして、データセット名を指定する場合には、普通前後の修飾子を省略してユーザ指定名（一般名）のみを入力すれば良い。これを部分名指定という。例えば、ＥＤＩＴコマンドを

E ISHIKAWA(TARO1) NE FORT77(FIXED)　＜＋Ｅ＋＞

と入力すると、データセット名として AB9999.ISHIKAWA.FORT77のメンバTARO1を指定したことになる（EはEDITコマンドの簡略名、NEはNEWオペランドの簡略形）。

これに対して、データセットの完全修飾名を入力して指定する方法を完全修飾名指定と呼び、データセット名全体を一対の引用符″'″でくくって指定する。

（例）　'AB9999.KANAZAWA.DATA(CENTER1)'

なお上記の例のように区分データセット（詳細は第２章参照）のメンバを指定する場合は、データセット名の直後にメンバ名を括弧でくくって指定する。

# 第2章　　データセットの種類と属性

## 2－1　　データセット編成

ＴＳＳで取り扱うことのできる広義のデータセットの入出力装置としては、直接アクセス記憶装置（**DASD**；Ｄｉｒｅｃｔ　Ａｃｃｅｓｓ　Ｓｔｏｒａｇｅ　Ｄｅｖｉｃｅ，具体的には磁気ディスクパック装置などを指す）、磁気テープ装置、ユニットレコード装置（カード入力装置、カードパンチ機、ラインプリンタ）及び端末装置などを指定することができる（注１）。しかし、アクセス（読み出し）速度、データの保存の便利さなどの点からＤＡＳＤを用いることが断然有利である。

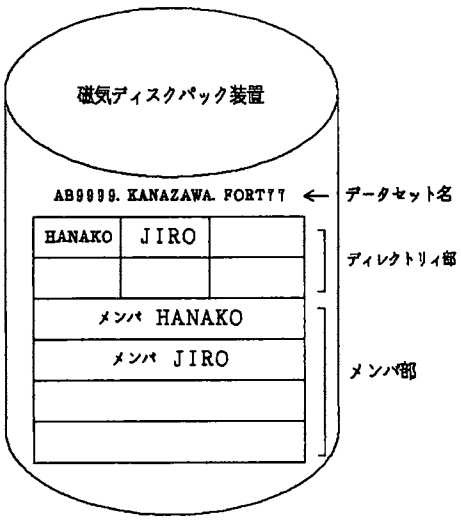



図3　区分データセットの構造

ＤＡＳＤ上に作成するデータセットには、順データセット（順編成データセット）と区分データセット（区分編成データセット）がある。順データセットとは１個のデータセットに対して確保された記憶域の最初から順にデータの入出力を行うデータセットをいう。これに対して、区分データセットとは１個のデータセットの中をいくつかの区域（メンバ）に分割し、その各区域内で順データセットと同様に順にデータの処理を行えるように作成される。区分データセットはＤＡＳＤ上のみに作成することが可能である。

区分データセットの構造を図３に示す。すなわち、区分データセットはディレクトリ部（索引部）とメンバ部とから構成され、ディレクトリ部にはメンバ名や各メンバの先頭アドレスなどが記憶されている。またメンバ部の各メンバにはソースプログラム、実行時のデータやロードモジュールなどのデータを順編成で格納することができる。

新しいデータセットをＥＤＩＴコマンドで確保する場合（オペランドとしてＮＥＷ指定）、位置オペランドであるデータセット名として、一般名のみを指定すれば順データセットが作成される。それに対して、一般名の直後に括弧付きでメンバ名も指定すれば区分データセットが作成される。１人のユーザに対してデータセット数は最大１０個までしか確保することが許されない（注２）から、多種

類のソースプログラム、データなどを保存する場合には区分データセットを用いるべきである。

（注１）当センタではＤＡＳＤ及び端末装置に限られる。

（注２）実際には１１個以上のデータセットを確保することが可能であるが翌日には１１個目以降は消去される。したがって誤って１１個以上作成した場合は整理して不要なデータセットを消去しなければならない。このときＵＣＦＬコマンドによって保存したいデータセット全てがＵＣＦリスト（ＵＣＦＬコマンドによって得られる課金情報用のデータセットリスト）に存在することを必ず確認されたい。詳細は利用の手引き〝ＴＳＳ実習編（中級用）―ＦＯＲＴＲＡＮユーザのためのジョブ制御文とデータセット―〟（山下　邦弘著）〔１０〕を参照されたい。

## ２－２　データセット属性

データセットの内容をアクセスするには、その記録のレコード形式（ＲＥＣＦＭ）、ブロック長（ＢＬＫＳＩＺＥ）及び論理レコード長（ＬＲＥＣＬ）が必要である。これらデータセットの性質をデータセットの属性という。レコードとはファイル（注３）を構成する情報のひとまとまりの単位であり、プログラム処理上論理的に取り扱い易い単位として定義する。それを論理レコード長と呼ぶ。例えばカードでは１レコード８０バイトである。また実際の入出力装置上では、スペース効率や、データ転送効率の観点から、連続する複数個のレコードをひとまとまりとして記録するが、この記録単位をブロック（物理レコード）、その長さをブロック長（バイト単位）と呼ぶ。

レコード形式は次の三種類に大別される。

（１）固定長形式（Ｆ；Ｆｉｘｅｄ）…全レコードの長さが一定。

（２）可変長形式（Ｖ；Ｖａｒｉａｂｌｅ）…レコードの長さはまちまちであり、各レコードの先頭にそのレコード長を表示する。

（３）不定形式（Ｕ；Ｕｎｄｅｆｉｎｅｄ）…レコード長が一定していなく、また可変長と異なって所定の位置にレコード長が記入されてもいない。実際に読み込んだ長さによって始めてそのレコード長を知ることができる。

またブロック化は固定長レコード形式および可変長レコード形式に対してのみ可能である。それらを固定長ブロック化レコード形式（ＦＢ形式）、可変長ブロック化レコード形式（ＶＢ形式）と呼ぶ。図４、図５にそれらと非ブロック化形式（Ｆ形式、Ｖ形式）とを対比して示す。つまり、ブロック化形式では複数個のレコードを１ブロックとして記録するのに対し、非ブロック化形式では１レコードが１ブロックとして扱われる。更に、図５中に示すように可変長形式では各レコードの長さが、まちまちであるため各レコードの先頭にレコード長を記述する４バイトの領域（ＲＤＷ）が必要であり、

また各ブロックの先頭にはブロック長を記述する４バイトの領域（ＢＤＷ）が必要となる。一方、固定長レコード形式には標準形式（S形式）が、また可変長レコード形式にはスパンド形式（S形式）というものもある。これらに関する詳細は〝データ管理機能説明書〟〔１１〕を参照されたい。

さてＥＤＩＴコマンドで新しくデータセットを確保すると、指定されたデータセットタイプによって上記データセット属性の標準値が自動的に決定される。それを表５に示す。また同表の指定値の範囲内で論理レコード長、ブロック長をＥＤＩＴコマンドのオペランドで指定することも可能である。

なお、データセットタイプがＯＢＪ、ＬＯＡＤのデータセットはそれらのレコード形式がそれぞれ、固定長形式（F形式）、不定形式（U形式）となる。

（注３）データセットとは、具体的にデータが磁気ディスクパック装置などに格納されている実体を指すのに対して、ファイルとは論理的なデータの集合体あるいは入出力先を指す。

表５　データセットタイプとデータセット属性　　単位バイト

| データセット属性 ＼ データセットタイプ | レコード形式 F／V | 論理レコード長　n 省略時 | 論理レコード長　n 指定値の制限 | ブロック長　m 省略値 | ブロック長　m 指定値の制限 | レコード中の行番号の位置 |
|---|---|---|---|---|---|---|
| FORT77 (FREE) | V | ２５５ | ―― | ３１２０ | m≦３１２０ | 始め８バイト |
| FORT77 (FIXED) | F | ８０ | n＝８０ | ３１２０ | m≦３１２０ | 終り８バイト |
| DATA | F | ８０ | n≦２５５ | ３１２０ | m≦３１２０ | 終り８バイト |
| CNTL | F | ８０ | n＝８０ | ３１２０ | m≦３１２０ | 終り８バイト |
| CLIST | F | ―― | n＝２５５ | ３１２０ | m≦３１２０ | 終り８バイト |
| | V | ２５５ | ―― | ３１２０ | m≦３１２０ | 始め８バイト |
| TEXT | F | ―― | n≦２５５ | ３１２０ | m≦３１２０ | 終り８バイト |
| | V | ２５５ | ―― | ３１２０ | m≦３１２０ | 始め８バイト |

（注）　レコード中の行番号の位置は、当然ＥＤＩＴコマンドのオペランド　ＮＯＮＮＵＭを指定しないでデータセットを確保した場合である。

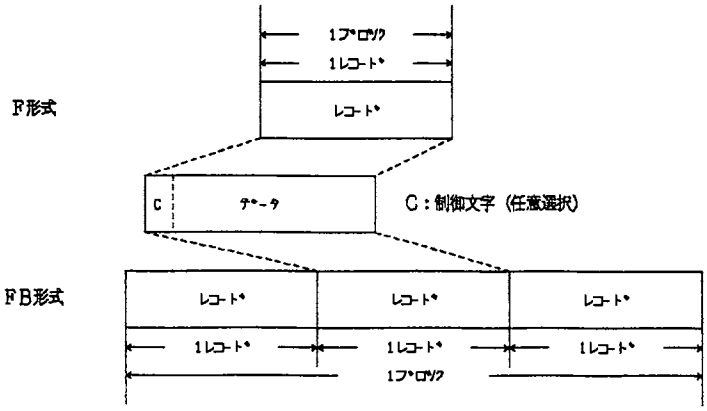



図4　固定長レコード形式

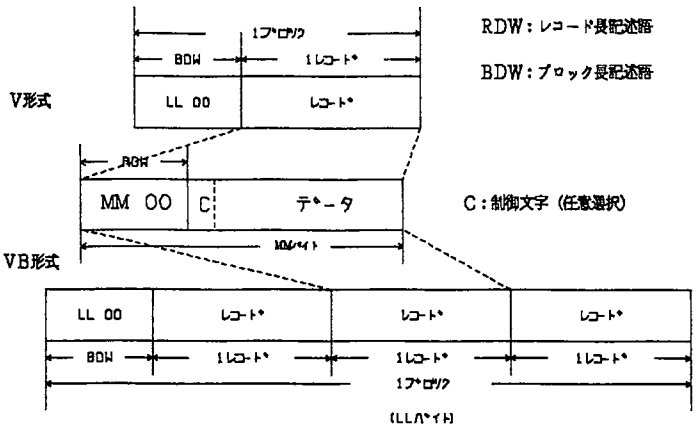



図5　可変長レコード形式

# 第3章　　データセット操作

本章では、データセット管理用コマンドについて説明する。前半はユーザが端末に向ってプログラムを直接実行する際にデータセットに作成したデータファイルからデータを入力したり、プログラム中のＷＲＩＴＥ命令による出力結果をデータセットに書き込んだりする場合に必要となるデータセットの割り当て、解放という操作について述べる。

後半では、データセットの内容の出力、複写、削除および他のデータセットの一部（又は全部）の内容を組み込んだりする各種のコマンドの使用法について説明する。

なお、データセットを使用してプログラムやデータなどの修正をくり返していくとデータセットがいっぱいになって修正した結果を保存できなくなる場合がある。これを解消するためにはデータセットの圧縮という操作が必要である。これに関しては利用の手引き〔１０〕を参照されたい。

## ３－１　　データセットの割り当て、解放とは

データセットの割り当て（ＡＬＬＯＣＡＴＥ）とは、ユーザがプログラム処理を行なう場合に必要な入出力ファイルのＤＤ名とデータセットとを結び付けることを意味する。ＤＤ名とは、作業に用いるファイルのデータ定義文（ＤＤ文）を識別するためにＤＤ文に付けられる名前であって、フォートランプログラムの処理過程では表６に示すものを用いる（詳細は〝ＦＯＲＴＲＡＮ７７使用手引書〟〔１２〕参照）。

表６　　フォートランプログラムの処理過程で用いるデータセット

| ＤＤ名 | データセットの内容 | 使用処理過程 |
|---|---|---|
| SYSIN | 原始プログラム | 翻訳 |
| SYSINC | 原始プログラム | 翻訳 |
| SYSPRINT | 印刷情報 | 翻訳、結合編集 |
| SYSTERM | 印刷情報 | 翻訳、結合編集 |
| SYSUT1 | 作業データ | 翻訳、結合編集 |
| SYSLIN | オブジェクトモジュール | 翻訳、結合編集 |
| SYSLIB | 自動呼出しライブラリ | 結合編集 |

表6　フォートランプログラムの処理過程で用いるデータセット

| | | |
|---|---|---|
| SYSLMOD | ロードモジュール | 結合編集 |
| 任　意 | ロードモジュール、他 | 結合編集 |
| SYSLOUT | 印刷情報 | 結合編集 |
| FT99F001 | 利用者の定義によるもの | 実　行 |

すなわち、DD名はシステムの定めた特定の文字列（たとえば"ＳＹＳＩＮ"、"ＳＹＳＰＲＩＮＴ"、…など）からなるものと、利用者が定義し、実行時の入出力に必要な"FT99F001"形式のものとがある。ＴＳＳユーザは通常、後者のＦＴ９９Ｆ００１形式のＤＤ名とその使い方について理解すれば十分である。なお、ＤＤ名FT99F001の99はデータセット識別名（論理機番）であり、００～９９の値をとることができる。一方、001はシーケンシャル番号であり、００１～９９９の値をとることができる。例えば、図6に示すようにＲＥＡＤ（5，…）、ＷＲＩＴＥ（6，…）命令に対するＤＤ名はそれぞれFT05F001、FT06F001である。

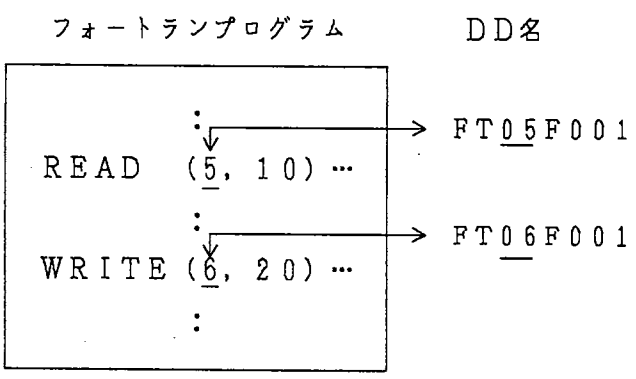



図6　入出力命令とＤＤ名の対応

さて、ＴＳＳコマンドによってＦＯＲＴＲＡＮ７７コンパイラの起動からプログラムの実行までを直接行うには、いく通りかの方法がある。これらの方法はプロンプタというものを介して行う方法と、プロンプタを介さない方法とに大別される。

プロンプタを介する方法には、表7に示す各種のコマンド（又はサブコマンド）がある。これらのコマンドでは、プロンプタによってコンパイルや結合編集に必要な各種の作業用データセットやオブジェクトモジュール、ロードモジュールを格納するデータセットなどが自動的に確保（割り当て）される。さらに、RUNコマンド、ＥＤＩＴのRUNサブコマンド、FORT7CLGコマンドなどで実行するときには、実行時の標準的な入出力ファイルのＤＤ名FT05F001、FT06F001は自動的に端末に割り当てられる。したがって、これらのコマンドを使ってプログラムを実行する場合は、入力データを端末のキーボードから入力し、出力結果を端末のディスプレイ画面（プリンタ型端末ではプリンタ）に出力する限り、データセットの割り当てというユーザにとってわずらわしい操作を全く必要としない。しかし、上記二つ以外のファイルを使用する場合やＲＥＡＤ（5，

…)、WRITE(6, …) 命令に対する入出力データをデータセットに確保する場合には上述のデータセットの割り当て操作が必要となってくる。

表7　プロンプタを介するコマンド

| コマンド名 | | 機能概略 |
|---|---|---|
| RUN | R | 翻訳・結合編集・実行を一つのコマンドで行う。 |
| RUN | E | 翻訳・結合編集・実行を一つのコマンドで行う。 |
| FORT77 | R | 翻訳、又はGOオプションによる実行を行う。 |
| LOADGO | R | 結合編集・実行を一つのコマンドで行う。 |
| LINK | R | 結合編集のみを行う。 |
| FORT7CLG | * | 翻訳・結合編集・実行を一つのコマンドで行う。オブジェクトモジュール、ロードモジュールの保存、実行も容易。 |

(注) Rはレディモードのコマンド、Eはエディトモードのサブコマンド、*はセンタコマンド

一方、FORTRAN77コンパイラの起動からプログラムの実行までをプロンプタを介さないで実行する方法として、CALLコマンドを用いる方法がある。この場合、フォートランプログラムの各種処理に対して表6に掲げたような種々のDD名を必要に応じて前もって割り当てる必要があり、ユーザにとっては面倒である。しかし、完成したプログラムをFORT77、LINKコマンドを用いて即時実行可能なロードモジュールまでおとしておき、それをCALLコマンドで実行する分にはDD名FT05F001、FT06F001は自動的に端末に割り当てられる故、面倒な操作は必要ない。したがって実行時の入力データを種々かえて実行したいユーザにとっては、ロードモジュールを一担保存しておけばそれをCALLコマンドで呼び出して直接実行できるのでCALLコマンドは有用である。

割り当てられたデータセットとDD名の接続関係を断ち切ることをデータセットを解放すると言う。この操作は現在割り当て中のDD名を他のデータセットに割り当てたい場合に必要となる。

## 3－2　割り当て、解放に関するコマンドの説明

データセットの割り当てにはALLOCATEコマンドおよびEDITのALLOCATEサブコマンドを用いる。また割り当てられたデータセットあるいはDD名を解放するにはFREEコマンドあるいはEDITのFREEサブコマンドを用いる。さらに現在割り当て中のデータセット名一覧のリスト

を出力するにはLISTALCコマンドがある。

（1）ALLOCATEコマンド（及びALLOCATEサブコマンド）

ALLOCATEコマンドは、ユーザが処理に必要なデータセットをセション開設後必要な時点で割り当てるために用いる。オペランドとしてデータセット名とDD名を対で指定し、データセットとDD名を結び付けるという形式で用いる。オペランドで指定するデータセットは、新規に作成するものでも、既存のものでも良い。またデータセットの存在するボリュームはDASDでもMTでもよく、ユニットレコード装置、端末、SYSOUTを指定することも可能である（注1）。さらにデータセットを連結することも可能である。それらに関する詳細は〝TSSコマンド文法書〟〔4〕を参照されたい。

ALLOCATEコマンドは、コマンドモード（〝READY〟を出力）で入力することができるが、その入力形式を次に示す（EDITのサブコマンドモードで入力可能なALLOCATEサブコマンドも全く同様である）。

ALLOCATE FILE(DD名) DATASET(データセット名)　　(注2)

・DD名はプログラム処理に必要なファイルを表わす。
・データセット名に星印〝＊〟を指定すると、データセットとして端末が割り当てられる。
・FILEオペランドとDATASETオペランドを入れ替えても良い。
・データセット名には内容識別修飾子まで、また区分データセットの場合はさらにメンバ名までを指定する必要がある。

（注1）当センタではDASDと端末のみ指定可能である。
（注2）太文字部はコマンド名、オペランドの簡略形（以下同様）。

（例1）　READ（5，…）の入力データを既存データセットKINDAI.DATA(HANAKO)　から読み込む場合

```
ALLOC F(FT05F001) DA(KINDAI.DATA(HANAKO)) <+E+>
```

（例2）　WRITE（6，…）の出力データを端末に出力する場合

```
ALLOC F(FT06F001) DA (*) <+E+>
```

(例3)　既存の入力データセットKANAZAWA.DATAを複数のジョブで同時に共用する場合

```
ALLOC F(FT05F001) DA(KANAZAWA.DATA) SHR <+E+>
```

オペランドSHRの替りにOLDを指定するとデータセットを専用することになる。すなわちＯＬＤ指定をしたデータセットは、それが解放されるまで他のジョブで用いることができない。既存データセットを拡張（追加書き）する場合はMODを、また新規にデータセットを作成する場合はNEWを指定する必要がある。

(例4)　ＷＲＩＴＥ（6，…）の出力データを新規の順データセットISHIKAWA.OUTLISTに保存する場合

```
ALLOC F(FT06F001) DA(ISHIKAWA.OUTLIST) NEW
      SP(2 2) T <+E+>
```

新規にデータセットを確保する場合には、NEWオペランドを指定し、磁気ディスクパック装置内での領域確保のためのオペランド（スペース量（SP）の初期割り当て量、増分量およびその単位（T：トラック））を指定する。上記の例では初期量２トラック、増分量２トラックである。（増分は１５回まで自動的にその指定したスペース量がとられる）。なお作成するデータセットが区分データセットである場合はディレクトリブロックの数（１単位当り２５６バイト）をDIオペランドで必ず指定する（例えばTオペランドに続いてDI(5)と入力する）。１単位当り１０～１５個のメンバ情報を格納できる。さらに、ＡＬＬＯＣＡＴＥコマンドで割り当てたすべてのデータセットは次に述べるＦＲＥＥコマンドによって解放しない限り、ＬＯＧＯＦＦコマンドによってセションを終了するまで有効である。

(2)　ＦＲＥＥコマンド（及びＦＲＥＥサブコマンド）

FREEコマンド及びＥＤＩＴのFREEサブコマンドはセションの途中で割り当て不要になったデータセットを解放する際に使用する。ＦＲＥＥコマンドはオペランドで、データセット名あるいはファイルのＤＤ名を指定する。その入力形式を次に示す。ＦＲＥＥサブコマンドも同様である。

```
  FREE DATASET(データセット名,…)
又は
  FREE FILE(DD名,…)
```

・データセット名は、内容識別修飾子までを指定する必要がある。区分データセットの場合はメンバ名も指定できる。

・データセット名を複数個指定すると、その一つ一つに対して解放処理がなされる。ＤＤ名についても同様である。

・端末装置を解放する場合は必ずＤＤ名指定によって行う。

（例１）　データセット名KINDAI.DATA(HANAKO)を指定して解放する場合

```
FREE DA(KINDAI.DATA(HANAKO))  <+E+>
```

（例２）　ファイル名FT05F001を指定して解放する場合

```
FREE F(FT05F001)  <+E+>
```

なお、ＡＬＬＯＣＡＴＥコマンドによって割り当てようとしたときに、ＦＩＬＥオペランドで指定したＤＤ名が、既に他のデータセットに割り当てられている場合、システムはそのＤＤ名に割り当てられているデータセットを解放する必要があるかどうかを端末ユーザに問い合せてくる。このとき〝FREE〟とキーボードから入力すれば、そのＤＤ名に先に割り当てられていたデータセットを解放し、新たに指定したデータセットが割り当てられる。〝END〟と入力すると、入力したＡＬＬＯＣＡＴＥコマンドの割り当て要求は取り消される。その例を次に示す。

（例３）

```
 READY
ALLOC F(FT05F001) DA(*) <+E+>
  :
  :
ALLOC F(FT05F001) DA(KINDAI.DATA(HANAKO)) <+E+>
 DATA SET AB9999.KINDAI.DATA NOT ALLOCATED,FILE IN USE
 ENTER 'FREE' OR 'END' +
FREE <+E+>
 READY
```

（３）　ＬＩＳＴＡＬＣコマンド

LISTALCコマンドは現在ユーザに割り当てられているデータセット名の一覧を出力する。この

一覧にはＬＯＧＯＮ時に自動的に定義されたデータセットやＡＬＬＯＣＡＴＥコマンドによって割り当てたデータセットを含む。入力形式は次のとおりである。

ＬＩＳＴＡＬＣ　〔ＳＴＡＴＵＳ〕　　（注３）

・オペランドを省略してコマンド名のみを入力すると割り当てられているデータセット名の一覧だけが出力される。

・オペランドＳＴも入力すると、データセット名に対応したＤＤ名とデータセットの状態（ＮＥＷ、ＭＯＤ、ＯＬＤ、ＳＨＲ）および解放後の処置（注４）（ＤＩＳＰ情報）を出力する。

（注３）コマンドの入力形式を記述する場合の約束として、小括弧（　）はそのままタイプ入力する。中括弧｛　｝はたとえば｛ＦＯＲＴ７７　ＣＮＴＬ　ＤＡＴＡ｝のように用いてその中から必要なオペランドを選択することを意味する。一方、大括弧〔　〕はその中のオペランドが省略可能であることを意味する。

（注４）データセットに対する処置には次の四つの種類がある。

・ＣＡＴＬＧ…データセットをカタログし、解放後も保存する（カタログの説明は３－４（１）を参照されたい）。

・ＫＥＥＰ…データセットを解放後も保存する。

・ＵＮＣＡＴＬＧ…データセットを解放後、カタログよりはずして保存する。

・ＤＥＬＥＴＥ…データセットを解放後削除する。

解放後の処置についてはＡＬＬＯＣＡＴＥコマンドで陽に指定することも可能であり、またＦＲＥＥコマンドによってそれを変更することも可能であるが詳細は〝ＴＳＳコマンド文法書〟〔４〕を参照されたい。なお、ＥＤＩＴコマンド、ＡＬＬＯＣＡＴＥコマンドによって新規に作成したデータセットは通常カタログされる。

使用例を図７に示す。ＳＹＳＯＵＴはＳＹＳＯＵＴデータセット、ＴＥＲＭＦＩＬＥは端末装置（ついでながらＳＹＳＩＮはＳＹＳＩＮデータセット）を意味する。図中の井印で示すように、ＳＹＳＩＮ、ＳＹＳＯＵＴ、ＴＥＡＭＦＩＬＥ（端末）データセットの場合は、データセットの状態、処置は出力されない。

なお、図中の井印は説明のために付けたものであり、実際には出力されない（以下同様とする）。

## ３－３　データセット割り当て、解放コマンドの使用例

図８は、入力データを区分データセットKANAZAWA.DATA(JIRO)に作成し、区分データセットKANAZAWA.FORT77(TARO)にある四則演算プログラムをＲＵＮサブコマンド（４－２節参照）で実行する例である。ただし出力結果は新規の順データセットKANAZAWA.OUTLISTに出力するものとする。

```
READY
LISTA ST <+E+>
 --DDNAME---DISP--
 SYS1.TGSPLIB
   STEPLIB   SHR,KEEP
 SYS1.CMDPROC
   SYSPROC   SHR,KEEP
 AICS.CMDPROC
             SHR,KEEP
 SYS1.HELP
   SYSHELP   SHR,KEEP
#  TERMFILE  SYSPRINT
#  TERMFILE  SYSIN
#  TERMFILE  SYSTSPRT
#  TERMFILE  SYSTSIN
#  SYSOUT    INTRDR
#  SYSOUT    CDFILE
 APP1.MACRO
   MACRO     SHR,KEEP
 APPF.LINKLIB
   LKLDSLIB SHR,KEEP
#  TERMFILE  FT05F001
#  TERMFILE  FT06F001
 READY
```

図７　ＬＩＳＴＡＬＣコマンドの使用例

＃１はデータセットＫＡＮＡＺＡＷＡ．ＤＡＴＡ（ＪＩＲＯ）を新規に作成している。オペランドのＮＥＷ指定により、ＥＤＩＴのＩＮＰＵＴモードとなり行番号が自動的に出力されてくるので、＃７のＲＥＡＤ命令のデータＡ、Ｂをプログラム中のフリーフォーマット指定（注４）に従って、コンマ、（又はブランク）で区切って順次入力する。入力データがつきると何も入力しないで、出力されてきた行番号に引き続いてＥＮＴＥＲキーを押す（＃３）とＥＤＩＴのサブコマンドモードとなり、〝Ｅ〟が出力される。＃４でエデットモードを終了し、入力テキストを保存する。＃５でＫＡＮＡＺＡＷＡ．ＦＯＲＴ７７（ＴＡＲＯ）のソースプログラムを編集用データセットに呼び出す。ＥＤＩＴのＬＩＳＴサブコマンドでプログラムリストを出力してみる（＃６）。

＃９はＡＬＬＯＣＡＴＥサブコマンドによってデータセットＫＡＮＡＺＡＷＡ．ＤＡＴＡ（ＪＩＲＯ）をＤＤ名ＦＴ０５Ｆ００１に割り当てている。＃１０は同様に新規の出力用順データセットＫＡＮＡＺＡＷＡ．ＯＵＴＬＩＳＴをＤＤ名ＦＴ０６Ｆ００１に割り当てている。スペース量の指定は出力する量に応じて選択すべきである。＃１１のＲＵＮサブコマンドによってプログラムを実行し、終了後＃１２のＤＳＰＲＩＮＴコマンドによってプリンタに出力した結果を図９に示す。さらに＃１３、＃１４で＃９、＃１０の割り当てを解放している。

（注４）ＲＥＡＤ、ＷＲＩＴＥ文のフォーマット文の文番号の代りに星印〝＊〟を書くと、ＲＥＡＤ命令に従ったデータを必要な個数だけコンマ又はブランクで区切って並べれば良い。実数はＦタイプでもＥタイプでも良い。又、ＷＲＩＴＥ命令では適当な書式で出力してくれる（＃８）。

```
      READY
#1    E KANAZAWA(JIRO) NE DATA <+E+>
      INPUT
#2    00010 1.,2. <+E+>
      00020 5.6,8. <+E+>
      00030 13.6,100. <+E+>
      00040 1.0E+3,111. <+E+>
#3    00050 <+E+>
      E
#4    END S <+E+>
      SAVED IN DATA SET 'AB9999.KANAZAWA.DATA(JIRO)'
      READY
#5    E KANAZAWA(TARO) FORT77(FIXED) <+E+>
      E
#6    L <+E+>
#7    00010   110 READ(5,*,END=120) A,B
      00020       WA=A+B
      00030       SA=A-B
      00040       SEKI=A*B
      00050       SYO=A/B
#8    00060       WRITE(6,*) A,B,WA,SA,SEKI,SYO
      00070       GO TO 110
      00080   120 STOP
      00090       END
      END OF DATA SET
#9    ALLOC F(FT05F001) DA(KANAZAWA.DATA(JIRO)) SH <+E+>
#10   ALLOC F(FT06F001) DA(KANAZAWA.OUTLIST) NE SP(2 2) T CA <+E+>
#11   RUN <+E+>
      FORTRAN77 COMPILER ENTERED
      END OF COMPILATION (OPTIMIZATION COMPILER)
      END OF GO, SEVERITY CODE=00
      E
      END <+E+>
#12   DSPRINT KANAZAWA.OUTLIST PC01 NON CCHAR <+E+>
      REQUEST QUEUED (*00025)
      READY
#13   FREE F(FT05F001) <+E+>
      READY
#14   FREE DA(KANAZAWA.OUTLIST) <+E+>
      READY
```

図8　ＡＬＬＯＣＣＡＴＥコマンド、ＦＲＥＥコマンドの使用例

## ３－４　その他のデータセット管理用コマンド

データセット管理用コマンドには表１に示したように、上記のＡＬＬＯＣＡＴＥコマンド、ＦＲＥＥコマンド、ＬＩＳＴＡＬＣコマンドの他にＬＩＳＴＣＡＴ、ＬＩＳＴＤＳ、ＣＯＰＹ、ＤＥＬＥＴ

```
AB9999    DATE 81.09.01    TIME 14:59:35  AB9999.KANAZAWA.OUTLIST

1.00000000 2.00000000 3.00000000 -1.00000000 2.00000000 0.500000000
5.60000038 8.00000000 13.6000004 -2.39999962 44.8000031 0.700000048
13.6000004 100.000000 113.599991 -86.3999939 1360.00000 0.135999978
1000.00000 111.000000 1111.00000 889.000000 111000.000 9.00900841
```

図9　図8の出力をＤＳＰＲＩＮＴコマンドで取り出した結果

Ｅ、ＲＥＮＡＭＥ、ＣＯＮＤ、ＣＯＭＰ、ＬＩＳＴＶ、ＵＣＦＬなどの各種コマンドがある。本節ではこれらのコマンドのうち、データセット名や区分データセットのメンバ名の一覧を出力するＬＩＳＴＣＡＴコマンド、ＬＩＳＴＤＳコマンド、データセットの複写、削除、名前の変更などに関するＣＯＰＹコマンド、ＤＥＬＥＴＥコマンド、ＲＥＮＡＭＥコマンド及びデータセットの組み込み用のＭＥＲＧＥコマンド（このコマンドは表１の分類では編集用コマンドに属する）について、その使用法を解説する。

（1）データセット名一覧の出力

ユーザが作成した、カタログされているデータセット名の一覧を出力するには、ＬＩＳＴＣＡＴコマンドを用いる。通常、正規に作成したデータセットでは、その名前、ボリューム通番、装置タイプなどに関する情報がシステムのカタログデータセットの登録簿（カタログ）に登録されており、データセットの管理に用いられる。コマンドの入力形式は次のとおり。

```
READY
LISTC <+E+>
 IN CATALOG: SYS1.VSAMCTLG
 AB9999.ABC.OUTLIST
 AB9999.ISHIKAWA.DATA
 AB9999.KANAZAWA.DATA
 AB9999.KANAZAWA.FORT77
 AB9999.KINDAI.OBJ
 AB9999.TOYAMA.LOAD
READY
```

図１０　ＬＩＳＴＣコマンドの使用例

```
LISTCAT
```

使用例を図１０に示す。

（2）　データセットの属性、メンバ名一覧の出力

作成したデータセットが格納されているボリューム通番、属性、データセット編成および区分データセットのメンバ名の一覧を出力するには、LISTDSコマンドを用いる。入力形式は次のとおり

LISTDS （データセット名リスト）〔MEMBERS〕

・データセット名リストには複数個のデータセット名をブランク（又はコンマ）で区切って指定する。1個のときは括弧を省略できる。

・MEMBERSオペランドを指定すると、区分データセットのメンバ名の一覧を出力する。

```
READY
LISTDS KANAZAWA.FORT77 M <+E+>
AB9999.KANAZAWA.FORT77
--RECFM-LRECL-BLKSIZE-DSORG
  FB     80     3120     PO
--VOLUMES--
  USER02
--MEMBERS--
  HANAKO
  JIRO
  TARO
READY
```

図11　LISTDSコマンドの使用例

使用例を図11に示す。この例ではデータセット名KANAZAWA.FORT77の存在するボリューム通番がUSER02であり、レコード形式がFB形式、論理レコード長が80バイト、ブロックサイズが3120バイトであること及びデータセット編成が区分編成（PO）であることを示している。またメンバはHANAKO、JIRO、TAROの三個が存在する。

なお、データセット編成の表示の種類にはPS（順編成）、PO（区分編成）、IS（索引順編成）、DA（直接編成）などがある。

（3）　データセットの複写

データセットの複写にはCOPYコマンドを用いる。COPYコマンドは、順データセットまたは区分データセットの一メンバを、順データセットまたは区分データセットの一メンバのいずれかに複写することができる。あるいは区分データセットの全体を、別の区分データセットに複写することも可能である。入力形式を次に示す。

COPY　入力データセット名　出力データセット名

・出力側が新データセット又は新メンバの場合はそれが作成される。既存の場合は内容が置き換えられる。
・入出力データセット名は同一であってはならない。したがってあるデータセット内の一メンバの複写を同一データセット内の別のメンバとすることはできない（この場合は後述のＭＥＲＧＥコマンドを用いる）。
・入出力データセット名の指定には内容識別修飾子も付ける。

なお、可変長スパンドレコード形式（ＶＢＳ形式）のデータセットは複写できない。

（例）区分データセットISHIKAWA.FORT77のメンバTAROを区分データセットKANAZAWA.CNTLの一メンバTARO1として複写する場合

```
 READY
COPY ISHIKAWA.FORT77(TARO) KANAZAWA.CNTL(TARO1)<+E+>
 READY
```

（4）　データセットの内容の組み込み

あるデータセットの一部あるいは全部の内容の複写を他のデータセットに組み込むには、MERGEコマンドあるいはＥＤＩＴのMERGEサブコマンドを用いる。ＭＥＲＧＥコマンドの入力形式は次のとおり。

ＭＥＲＧＥ　データセット名１〔行番号１〔行番号２〕〕　データセット名２〔行番号３〕
〔RENUM〕

・データセット名１はレコードを取り出す側の名前（区分データセットはメンバ名を含む）を指定する。行番号１、行番号２は取り出すレコードの始めと終りのレコードの行番号であり、省略すると全レコードが取り出される。
・データセット名２は、レコードを挿入する側の名前を指定する。行番号３で指定したレコードの直後に挿入される。これを省略すると、データセットの最終レコードの後に追加される。
・ＲＥＮを指定すると組み込み処理の終了後、データセット名２の行番号は付け直される。

なおＥＤＩＴのＭＥＲＧＥサブコマンドの場合は、挿入されるデータセット名２は当然編集中のデ

ータセットである故、指定する必要はない。また行番号3を省略すると、行指標（ＦＳ画面に表示されている最上行）の次に組み込まれる。さらにＲＥＮＵＭオペランドの入力の必要はなく、組み込み処理終了後、必ず自動的に行番号は付け直される。

（例）　ＭＥＲＧＥサブコマンドの使用例を示す。

図１２（ａ）は既存の区分データセットKANAZAWA.FORT77(JIRO)のソースリストである。（ｂ）図は編集中の区分データセットKANAZAWA.FORT77(SABURO)のＭＥＲＧＥサブコマンド入力前のソースリストを示す。次のようにＭＥＲＧＥサブコマンドを入力すると、（ｃ）図のようにソースリストは変化し、行番号は付け直される。

```
E
M KANAZAWA.FORT77(JIRO)
 40 70 30  <+E+>
E
```

```
      0010        ******
      0020        -----
      0030        11111
      0040        22222
      0050        33333
      0060        44444
      0070        55555
      0080        66666
      0090        %%%%%
      0100        &&&&&
(A)  KANAZAWA.FORT77(JIRO)

      0010        AAAAAAA
      0020        BBBBBBB
      0030        CCCCCCC
      0040        DDDDDDD
      0050        EEEEEEE
      0060        FFFFFFF
(B)  KANAZAWA.FORT77(SABURO)
     (MERGE 前)

      0010        AAAAAAA
      0020        BBBBBBB
      0030        CCCCCCC
      0040        22222
      0050        33333
      0060        44444
      0070        55555
      0080        DDDDDDD
      0090        EEEEEEE
      0100        FFFFFFF
(C)  KANAZAWA.FORT77(SABURO)
     (MERGE 後)
```

図１２　ＭＥＲＧＥサブコマンドの使用例

（５）　データセットの消去

データセットの消去にはDELETEコマンドを用いる。入力形式は次のとおり。

DELETE （データセット名リスト）

・データセット名リストには複数個のデータセット名を記述できる。指定個数が１個の場合は括弧を省略できる。

・区分データセットではメンバ名も指定すると、指定したメンバのみを削除することができる。区分データセットのデータセット名のみを指定するとそのデータセット（全てのメンバ）が削除される。

(例1) 区分データセットISHIKAWA.FORT77のすべてのメンバを削除する場合

```
READY
DEL ISHIKAWA.FORT77 <+E+>
ENTRY(A) AB9999.ISHIKAWA.FORT77 DELETED
READY
```

(例2) 区分データセットISHIKAWA.FORT77のメンバJIROを削除する場合

```
READY
DEL ISHIKAWA(JIRO) <+E+>
DATA SET AB9999.ISHIKAWA HAS FOLLOWING QUALIFIERS
CNTL        DATA          FORT77
ENTER QUALIFIER FROM ABOVE LIST + FORT77 <+E+>
MEMBER JIRO DELETED
READY
```

上記の例のように同一のユーザ指定名を持つデータセットがいくつか存在する（上記の例ではISHIKAWA.CNTL、ISHIKAWA.DATA、ISHIKAWA.FORT77）にもかかわらず、内容識別修飾子の入力を省略すると、システムは判別できないため内容識別修飾子の入力を要求してくる。そのときには内容識別修飾子（FORT77、CNTL、DATAなど）のみを例のように入力すれば良い。

(6) データセット名、メンバ名の変更

データセット名や区分データセットのメンバ名を変更したい場合には、RENAMEコマンドを用いる。入力形式は次のとおり。

```
RENAME 旧データセット名 新データセット名
・メンバ名を変更する場合には括弧付きで指定する。
```

(例1) データセット名ISHIKAWA.FORT77を TOYAMA.CNTLに変更する場合

```
READY
REN ISHIKAWA.FORT77 TOYAMA.CNTL <+E+>
```

READY

(例2) データセット名ISHIKAWA.FORT77のメンバ名JIROをSABUROに変更する場合

READY

REN ISHIKAWA.FORT77(JIRO) (SABURO) <+E+>

READY

上記の例のように新旧データセット名が同一の場合は新データセット名の指定を省略できる。内容識別修飾子を省略した場合は、DELETEコマンド同様、入力を要求してくる。

# 第4章　　実行方法のいろいろ

ＴＳＳによってプログラムを実行する方法には、（1）ＳＵＢＭＩＴコマンド（又はサブコマンド）によってリモートバッチジョブを依頼する方法、（2）ＲＵＮコマンドを始めとする種々のコマンドによって直接実行する方法（ＪＣＬ文を必要としない）、（3）会話的にディバッグを進めるインタラクティブディバッグ（詳細は利用の手引き〔8〕参照）の三種類がある。本章では、まず方法（2）によるフォトランプログラムの実行までの処理過程とプログラム処理用コマンドの関係を概説する。そしてその各種コマンドの文法を実例をおりまぜて解説する。更にソースプログラム、ＪＣＬ文、実行時の入力データが別々のデータセットにある場合の実行方法（1）の例も示す。

## 4－1　　翻訳、結合編集、実行とコマンドとの対応

フォートランのソースプログラム（データセットタイプが"FORT77(FIXED)"）を実行するまでには、翻訳、結合編集、実行の三つの処理が必要である。まずフォートランのソースプログラムはＦＯＲＴＲＡＮ７７コンパイラにより翻訳されてアセンブラ形式のオブジェクトモジュールが作成される。次にリンケージエディタによって結合編集され、機械語によるロードモジュールが作成される。そのロードモジュールは、配列の領域確保やフォートランの組み込み関数などを結合する以前のリロケータブルな相対形式とそれらの処理を済ませたアブソリュートな実行形式とに分類される。相対形式はサブルーチン単位にプログラムを保存することが可能であり、それら相対形式のサブルーチンは他のプログラムを実行する際にも結合することができる（主プログラムも相対形式で保存することが可能であるが、ソースプログラムの先頭にＰＲＯＧＲＡＭ文が必要である）。一方、実行形式は主プログラム、サブルーチンプログラム、配列の領域など全てをひとまとまりとして１つのメンバ名を付けてデータセットに保存するので、それをＣＡＬＬコマン

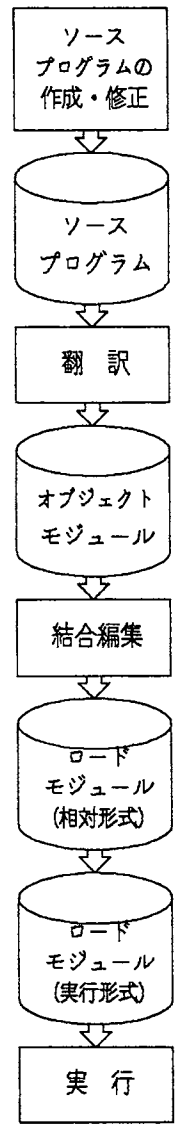



図１３　プログラムの処理過程

ドで呼び出すことによって即時実行が可能である。なお、相対形式は実行形式に比較して保存のための領域はかなり少なくてすむ。翻訳から実行までの過程を図１３に示す。

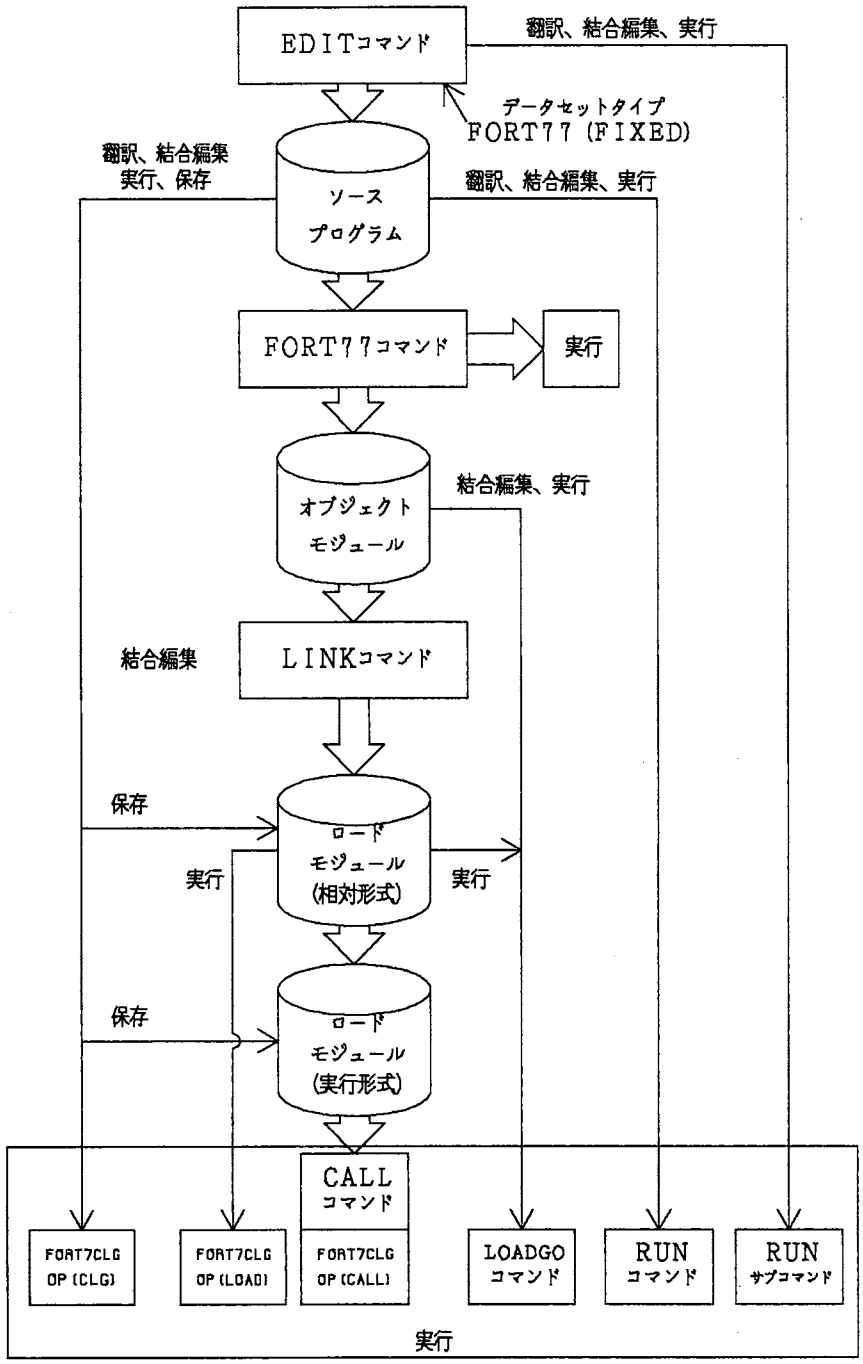



図１４　プログラム処理過程とそれに対応するコマンド

利用者の立場から見れば、完成したプログラムを何度も実行するつもりならばロードモジュールにおとしておく方が、翻訳、結合編集に要するＣＰＵタイムを節約でき有利である。特に実行時のデータのみを替えて何度も実行する場合は、実行形式でプログラムを保存しておくと良い。また、いくつもの異なったプログラムで共通に利用したいサブルーチンなどは相対形式で保存しておき、各プログラムの実行の際に結合すると良い。同様に、たとえ開発中のプログラムであっても完成したサブルーチンは相対形式で保存しておき、それを結合してディバッグを進めることも可能である。ただし、ブロックデータ文は実行文を含まないためにロードモジュールにおとしておくことはできないので注意されたい。

それぞれのモジュールとそれに対応するコマンドとの関係を図１４に示す。簡単な短いプログラムは、RUNコマンドやＥＤＩＴのRUNサブコマンドを用いて一気に実行した方が手軽である。更に、ＲＵＮサブコマンドはプログラムの編集（エデット）中に実行できるので、文法エラーのチェックに最適である。しかし、プログラムが大きくなると作業領域が不足してそれらのコマンドでは実行不可能となる場合がある。その場合にはFORT77コマンドあるいはFORT7CLGコマンドによらなければならない。一方、ロードモジュールを保存するにはFORT77→LINKコマンドによるかあるいはFORT7CLGコマンドによらなければならない。また、ロードモジュールのプログラムを実行するにはCALLコマンドあるいはFORT7CLGコマンドがある。次節以降に、これらの種々の実例を示し、コマンドの使用法その他の詳細を解説する。

## ４－２　ＲＵＮコマンド及びＲＵＮサブコマンド

RUNコマンド及びＥＤＩＴのRUNサブコマンドは、データセットタイプＦＯＲＴ７７（ＦＩＸＥＤ）のデータセットに作成したソースプログラムを最も簡単な手順で翻訳、結合編集、実行を行うことができる。特にＲＵＮサブコマンドは、編集用データセットに作成したソースプログラムをすぐに実行し、文法エラーをチェックするのに有用である。

### （１）　コマンド（及びサブコマンド）の説明

RUNコマンドの入力形式は次のとおりである。

```
RUN  データセット名(メンバ名)  FORT77  FIXED  〔LIB(区分データセット
                                名)〕
```

・翻訳、結合、実行しようとするフォートランソースプログラムが格納されているデータセット

名と、それにひき続きメンバ名を括弧でくくって指定する。

・相対形式で登録されているサブルーチンを結合して実行したい場合は、それが格納されている区分データセット名のみ（データセットタイプは必ず〝LOAD〟であること、タイプの入力は不要）をオペランドLIBで指定する。これをライブラリィ指定という。

ＥＤＩＴのデータセットタイプオペランドFORT77(FIXED)とは異なり、FORT77、FIXEDは別々のオペランドであるから、その入力に注意すること。

一方、ＥＤＩＴのサブコマンドモードで入力可能なRUNサブコマンドの入力形式を次に示す。

```
RUN 〔LIB(区分データセット名)〕
・ＬＩＢ指定の意味はＲＵＮコマンド同様。
```

ＲＵＮコマンドやＲＵＮサブコマンドを入力すると、ＤＤ名FT05F001およびFT06F001は自動的に端末に割り当てられる。ＲＵＮサブコマンドで実行し、入力データをデータセットから読み込み、出力結果をデータセットに保存する例を３－３節で既に示した。以下に二つの使用例を示す。なお、オブジェクトモジュールやロードモジュールを保存することはできない。

（２）　ＲＵＮコマンドの簡単な使用例

図２、図８で引用したデータセットKANAZAWA.FORT77のメンバ名TAROに保存されている四則演算のプログラムをＲＵＮコマンドによって実行する例を図１５に示す。

データをキーボードより入力し、結果を端末ディスプレイ画面に出力するものとする。＃１のようにＲＵＮコマンドを入力すると、ＦＯＲＴＲＡＮ７７のコンパイラが呼び出され、＃２のようにその由のメッセージが出力される。文法エラーがあるとこの後に出力され、処理は終了する。エラーがないと＃３のメッセージが出力され、すぐ実行に移される。プログラム中にＲＥＡＤ命令があると、＃４のようにその行番号と疑問符〝？〟が出力されて、端末はデータの入力待ちとなるので、必要なデータを入力する。データ数が不足の場合は疑問符を出力して再び入力を促してくる。またデータとして不正な文字列を入力すると、その由のエラーメッセージが出力され、再び入力を要求してくる。正しくデータを入力すると、プログラムは実行され＃５のようにＷＲＩＴＥ命令による出力結果が表示される。本プログラムのようにデータの終りをＲＥＡＤ文のＥＮＤ指示子で判定する場合は＃６のように〝/*〟を入力すると、データの終りをシステムが判定する。実行が終了すると、＃７のよう

にその由のメッセージと完了コードが出力される。なおＥＮＤ指示子のないＲＥＡＤ命令に対しても〝/*〟を入力することにより終了することができるが、この場合エラー表示がなされる。

```
      READY
#1    RUN KANAZAWA(TARO) FORT77 FIXED <+E+>
#2     FORTRAN 77 COMPILER ENTERED
#3     END OF COMPILATION (OPTIMIZATION COMPILER)
#4     00010 ?
      2.,15. <+E+>
#5     2.00000000 15.0000000 17.0000000 -13.0000000 30.0000000 0.133333325
       00010 ?
      2.5E+3,100. <+E+>
       2500.00000 100.000000 2600.00000 2400.00000 250000.000 25.0000000
       00010 ?
#6    /* <+E+>
#7     END OF GO,SEVERITY CODE=00
       READY
```

図１５　ＲＵＮコマンドの使用例

（３）　ライブラリィ指定の例

図１６は科学用サブルーチンライブラリィＳＳＬにある〝ＣＡＲＤＮＳ〟を用いて実係数三次方程式の根を求める例である。＃１～＃６はソースリストである。ＳＳＬを始めるとするセンタライブラリィを結合して実行する場合も、ユーザライブラリィと同様指定する必要がある（ＦＯＲＴ７ＣＬＧコマンド以外のコマンドで実行する場合も同様である）。参考までにＳＳＬを始めとする全てのライブラリィは'APPF.LINKLIB'に格納されている。自分のライブラリィ以外のライブラリィを指定する場合は、完全修飾名を引用符でくくって指定する。＃７のようにＲＵＮサブコマンドを入力すると、エディト中のデータセットタイプに対応したコンパイラが呼び出される。＃２のＲＥＡＤ文に対応して、＃８のように４つの係数を高次の方から順に入力すると、計算が実行され＃４、＃５に対応した出力結果が＃９、＃１０のように出力される。

## ４－３　ＦＯＲＴ７７、ＬＩＮＫ、ＣＡＬＬコマンド

FORT77→LINK→CALLコマンドを連続して用いると翻訳、結合編集、実行が可能である。その過程で生ずるオブジェクトモジュール、ロードモジュールを保存することができる。またＦＯＲＴ７７コマンドでＧＯオペランドを指定すると、直接実行することも可能である。

```
     E
    └<+E+>
#1   0010 C      ***** ROOTS OF 3-DEGREE EQUATION *****
     0020         REAL A(4),RP(3),IP(3)
#2   0030         READ(5,*)  (A(I),I=1,4)
     0040 C      **********************************
#3   0050         CALL CARDNS(A,RP,IP,ILL)
     0060 C      **********************************
#4   0070         WRITE(6,600)  (A(I),I=1,4)
     0080    600 FORMAT(/1H ,'(',E12.5,')*X**3+(',E12.5,
     0090       1')*X**2+(',E12.5,')*X+(',E12.5,')=0'/)
     0100         IF(ILL.EQ.0) GO TO 10
     0110         STOP
     0120      10 DO 20 I=1,3
#5   0130      20 WRITE(6,610)  I,RP(I),IP(I)
     0140    610 FORMAT(1H ,5X,'X',I2,' =('E12.5,')+(',
     0150       1E12.5,')*SQRT(-1)')
     0160         STOP
#6   0170         END
     ***** END OF DATA SET *****
     E
#7   RUN LIB('APPF.LINKLIB') <+E+>
     FORTRAN 77 COMPILER ENTERED
     END OF COMPILATION (OPTIMIZATION COMPILER)
     00030 ?
#8   1.,2.,3.,4. <+E+>
#9    ( 0.10000E+01)*X**3+( 0.20000E+01)*X**2+( 0.30000E+01)*X+( 0.40000E
     +)=0
#10        X 1 =(-0.16506E+01)+( 0.0         )*SQRT(-1)
           X 2 =(-0.17469E+00)+( 0.15469E+01)*SQRT(-1)
           X 3 =(-0.17469E+00)+(-0.15469E+01)*SQRT(-1)
     END OF GO.SEVERITY CODE=00
     E
```

図16　ライブラリィ指定の例

（1）　コマンドの説明

FORT77コマンドはフォートランのソースプログラムを翻訳してオブジェクトモジュールを作成する場合及び翻訳～実行を行う場合に用いる。その入力形式を次に示す。

FORT77　データセット名1（メンバ名）　〔OBJECT（データセット名2（メンバ名））〔NAME〕〕〔GO　〔LIB（区分データセット名）〕〕

・データセット名1（メンバ名）はソースプログラムが保存されているデータセットタイプがFORT77（FIXED）なるデータセット名とメンバ名を指定する。

・データセット名２（メンバ名）は作成したオブジェクトモジュールを保存する場合にそのデータセット名とメンバ名を指定する。名前は任意である。データセットタイプを省略するとOBJが自動的に付加される。

・NAMEオペランドはプログラムを翻訳し、ＬＩＮＫコマンドを用いて相対形式で登録したい場合に必ず指定する。

・GOオペランドは翻訳後、実行したい場合に指定する。このときオブジェクトモジュールは保存できない（OBJオペランドを指定してはならない）。

・LIBオペランドによるライブラリィ指定の方法は、ＲＵＮコマンドと同様である。ＧＯオペランドを指定した場合に限り指定可能である。

ＦＯＲＴ７７コマンドは上記の他に多数のオペランド指定が可能である。特にディバッグ用のDEBUG（{ARGCHK TRACE SUBTRACE INIT SUBCHK}）オペランドは有用である。詳細は〝ＦＯＲＴＲＡＮ７７使用手引書〟〔１２〕を参照されたい。

LINKコマンドはＦＯＲＴ７７コマンドによって作成されたオブジェクトモジュールを結合編集してロードモジュールを作成し、保存する時に用いる。ＬＩＮＫコマンドによってオブジェクトモジュールから直接実行することはできない（オブジェクトモジュールを直接実行にはLOADGOコマンドを用いるが、詳細は〝ＴＳＳコマンド文法書〟〔４〕を参照されたい）。

LINKコマンドの入力形式を次に示す。

LINK データセット名１（メンバ名） LOAD（データセット名２〔（メンバ名）〕）
〔NCAL〕 〔FORTLIB〕 〔LIB（区分データセット名）〕

・データセット名１（メンバ名）は結合編集しようとするオブジェクトモジュールが保存されているデータセット名とメンバ名を指定する。

・データセット名２は作成されたロードモジュールを出力する区分データセット名を指定する。名前は任意。実行形式で登録する場合はメンバ名も指定する。相対形式でサブルーチンを登録する場合はメンバ名を指定しない（サブルーチン名がメンバ名として登録される）。なおLO（ ）オペランドを省略すると、データセット名１（メンバ名）で指定したユーザ指定名、メンバ名でタイプLOADなるデータセットに保存される（相対形式で登録する場合はサブルーチン名がメンバ名として登録される）。

・オペランドNCALは相対形式で登録する場合のみ指定する。ただし、ＦＯＲＴ７７コマンドでNAMEオペランドを指定して翻訳されたオブジェクトモジュールに対してのみ有効である。

・ＦＯＲＴＬＩＢオペランドは実行形式でロードモジュールを登録する場合に指定する。

・LIBオペランドはライブラリィ指定を必要とする場合に指定する。詳細はＲＵＮコマンドの場合と同様。実行形式で登録する場合にのみ有効。

CALLコマンドは実行形式で登録されているプログラムを実行する場合に用いる。入力形式を以下に示す。

CALL　データセット名(メンバ名)

・データセット名(メンバ名)は実行しようとするプログラムが実行形式で格納されているデータセット名とメンバ名を指定する。

（2）　翻訳、結合編集、実行の例

まず、ＦＯＲＴ７７コマンドによってオブジェクトモジュールを作成し、ＬＩＮＫコマンドによりその保存されたオブジェクトモジュールから実行形式ロードモジュールを作成、保存する。更にＣＡＬＬコマンドによってそれを実行する例を示す。図１７に示すような最小二乗法による直線近似のソースプログラムがデータセット名　KANAZAWA.FORT77(LINEFIT)に格納されているものとする。

図１８は実行までの過程を示す。まず＃１のようにＦＯＲＴ７７コマンドを入力する。ソースプログラムの入力データセット名は勿論ＫＡＮＡＺＡＷＡ．ＦＯＲＴ７７でメンバ名はＬＩＮＥＦＩＴである。データセットタイプは入力の必要がない。作成されたオブジェクトモジュールを格納するデータセットとしてKANAZAWA.OBJ(LFOBJ)を指定したことになる。データセット名、メンバ名は任意である。例のようにソースと同一のデータセット名を指定しても、タイプが異なるためシステムは異なるデータセットとして取り扱う。念のためＲＥＡＤＹ出力後、＃２のようにＬＩＳＴＤＳコマンドによってＫＡＮＡＺＡＷＡ．ＯＢＪのメンバ名一覧を出力してみると＃３の如くメンバLFOBJは確かに存在する。次に＃４のようにＬＩＮＫコマンドを入力して実行形式ロードモジュールを作成する。オブジェクトモジュールの入力としてデータセットKANAZAWA.OBJ(LFOBJ)を指定したつもりだったが、データセットKANAZAWA.DATAが存在したため（システム上はデータセットタイプとして　ＯＢＪ、ＤＡＴＡとも入力対象として指定が可能）端末はデータセットタイプの入力待ちとなる。そこで＃６のようにタイプ〝OBJ〟を入力する。なお、＃４ではロードモジュールの出力データセットとしてＫＡＮＡＺＡＷＡ．ＬＯＡＤ（ＬＦＬＯＡＤ）を指定したことになる。処理が正常に終了すると＃７のようにメンバ名ＬＦＬＯＡＤが保存された由のメッセージが表

```
    AB9999  DATE 81.09.01 TIME 10:36:59  AB9999.KANAZAWA.FORT77(LINEFIT)

     0010 C      ***  LINE FITTING BY LEAST SQUARE METHOD  ***
     0020        DIMENSION DX(100),DY(100)
¥1   0030        READ(5,*)  N
¥2   0040        READ(5,*)  (DX(I),DY(I),I=1,N)
     0050        CALL LSQ(DX,DY,N,CA,CB)
¥3   0060        WRITE(6,600)  CA,CB
     0070   600 FORMAT(/1H ,'A =  ',F8.4,5X,'B =  ',F8.4/)
     0080        DO 10 I=1,N
     0090         ERR=DY(I)-(CA+CB*DX(I))
¥4   0100    10  WRITE(6,610)  I,DX(I),DY(I),ERR
     0110   610 FORMAT(1H ,5X,I5,3F10.3)
     0120        STOP
     0130        END
     0140 C      *****  SUBROUTINE PROGRAM FOR LINE FITTING  *****
     0150        SUBROUTINE LSQ(X,Y,N,A,B)
     0160        DIMENSION X(100),Y(100)
     0170        SX=0.
     0180        SY=0.
     0190        DO 10 I=1,N
     0200         SX=SX+X(I)
     0210    10  SY=SY+Y(I)
     0220        AN=N
     0230        XM=SX/AN
     0240        YM=SY/AN
     0250        SXX=0.
     0260        SXY=0.
     0270        DO 20 I=1,N
     0280         SXX=SXX+(X(I)-XM)**2
     0290    20  SXY=SXY+(X(I)-XM)*(Y(I)-YM)
     0300        B=SXY/SXX
     0310        A=YM-B*XM
     0320        RETURN
     0330        END
```

図１７　最小二乗法による直線近似

示される。また念のためＬＩＳＴＤＳコマンドによってＫＡＮＡＺＡＷＡ．ＬＯＡＤのメンバ名一覧を出力してみる（＃８）。最後に＃９のようにＣＡＬＬコマンドで実行形式プログラムを実行する。図１７の￥１、￥２のＲＥＡＤ文に対応して＃１０、＃１１のようにデータを入力すると、計算が実行され￥３、￥４に対応した出力＃１２、＃１３が画面に表示される。

実行終了後も、オブジェクトモジュールとロードモジュール（実行形式）がデータセットに保存されている。実行形式のプログラムを保存の後は、オブジェクトモジュールは不要であるのでＤＥＬＥＴＥコマンドで消去しておくべきである。

（３）相対形式の登録とライブラリィ指定による実行

```
     READY
#1   FORT77 KANAZAWA(LINEFIT) OBJ(KANAZAWA(LFOBJ)) <+E+>
     FORTRAN 77 COMPILER ENTERED
     END OF COMPILATION (OPTIMIZATION COMPILER)
     READY
#2   LISTD KANAZAWA.OBJ M <+E+>
     AB9999.KANAZAWA.OBJ
     --RECFM-LRECL-BLKSIZE-DSORG
       FB     80     80       PO
     --VOLUMES--
       USER04
     --MEMBERS--
#3     LFOBJ
     READY
#4   LINK KANAZAWA(LFOBJ) LOAD(KANAZAWA(LFLOAD)) FORTLIB <+E+>
#5   VALID TYPES FOR DATA SET KANAZAWA ARE OBJ AND DATA
     ENTER TYPE-
#6   OBJ <+E+>
#7    **MEMBER NAME** LFLOAD     NOW ADDED TO LIBRARY.
     READY
#8   LISTD KANAZAWA.LOAD M <+E+>
     AB9999.KANAZAWA.LOAD
     --RECFM-LRECL-BLKSIZE-DSORG
       U      **     13030    PO
     --VOLUMES--
       USER01
     --MEMBERS--
       LFLOAD
     READY
#9   CALL KANAZAWA(LFLOAD) <+E+>
     00030 ?
#10 9 <+E+>
     00040 ?
#11 1.,2.1,3.,6.3,4.,8.5 <+E+>
#12  A =  -0.0429      B =    2.1286
#13              1      1.000      2.100      0.014
                 2      3.000      6.300     -0.043
                 3      4.000      8.500      0.029
     READY
```

図１８　ＦＯＲＴ７７、ＬＩＮＫ、ＣＡＬＬコマンドによる翻訳、結合、実行の例

次に図１７に示した最小二乗法による直線近似プログラムのサブルーチンＬＳＱをいったん相対形式で登録し、それをライブラリィ指定してＦＯＲＴ７７コマンドで実行する場合の例を図１９に示す。ただし、図１７に示したソースリストのうち主プログラムはKANAZAWA.FORT77のメンバMAINに、またサブルーチンＬＳＱは同一データセット内のメンバLSQに保存されているものとする。

まず、図１９の＃１はＦＯＲＴ７７コマンドによってサブルーチンＬＳＱを翻訳するための入力を

```
      READY
#1    FORT77 KANAZAWA(LSQ) OBJ(KANAZAWA(LSQ)) NAME <+E+>
      FORTRAN 77 COMPILER ENTERED
      END OF COMPILATION (OPTIMIZATION COMPILER)
      READY
#2    LINK KANAZAWA.OBJ(LSQ) LO(KANAZAWA) NCAL <+E+>
      **MEMBER NAME** LSQ     WAS NOT FOUND BUT HAS BEEN ADDED TO LIBRARY.
      READY
#3    FORT77 KANAZAWA(MAIN) GO LIB(KANAZAWA) <+E+>
      FORTRAN 77 COMPILER ENTERED
      END OF COMPILATION (OPTIMIZATION COMPILER)
      00030 ?
#4    3 <+E+>
      00040 ?
#5    1.,2.1,3.,6.3,4.,8.5 <+E+>
#6     A =   -0.0429      B =    2.1286
#7                 1      1.000      2.100      0.014
                   2      3.000      6.300     -0.043
                   3      4.000      8.500      0.029
      END OF GO.SEVERITY CODE=00
      READY
```

図１９　ＦＯＲＴ７７コマンド、ＬＩＮＫコマンドによる相対形式プログラムの登録と実行

示す。相対形式で登録するための準備としてNAMEオペランドを入力する。作成されたオブジェクトモジュールはデータセットKANAZAWA.OBJ(LSQ)に保存される。文法エラーがあると、そのメッセージがこの後出力される。正常に終了した場合、続いて＃２のようにＬＩＮＫコマンドを入力する。相対形式で登録するため、NCALオペランドを指定し、ロードモジュールの出力データセット名のみを指定する。正常に終了すると、サブルーチン名ＬＳＱがメンバ名としてライブラリィに付加された由のメッセージが出力される。これで登録は完了である。

次に＃３のようにＦＯＲＴ７７コマンドによってデータセットKANAZAWA.FORT77(MAIN)に保存されている主プログラムを実行する。実行時のオペランドGOおよびライブラリィKANAZAWA.LOADの結合を指定する。ライブラリィ指定を忘れると、結合時に外部手続き名ＬＳＱが解決できない由のメッセージが出力され、処理は中止される。ＦＯＲＴ７７コマンド入力後、＃４～＃７のように実行される。なお入出力ファイルをデータセットに割り当てたい場合は、＃３の実行コマンドを入力する前にＡＬＬＯＣＡＴＥコマンドによって指定しておけば良い（３－３節参照）。

（４）　ライブラリィ結合による実行形式の登録と実行

図２０はKANAZAWA.FORT77(MAIN)の主プログラムを翻訳し、図１９の例で先に登録しておいた相対形式のサブルーチンＬＳＱ（KANAZAWA.LOAD(LSQ)）を結合して、実行形式

```
     READY
#1   FORT77 KANAZAWA(MAIN) OBJ(KANAZAWA(MAIN)) <+E+>
     FORTRAN 77 COMPILER ENTERED
     END OF COMPILATION (OPTIMIZATION COMPILER)
     READY
#2   LINK KANAZAWA(MAIN) LO(KANAZAWA(MAIN)) F LIB(KANAZAWA) <+E+>
     VALID TYPES FOR DATA SET KANAZAWA ARE OBJ AND DATA
     ENTER TYPE-
#3   OBJ <+E+>
     **MEMBER NAME** MAIN   WAS NOT FOUND BUT HAS BEEN ADDED TO LIBRARY.
     READY
#4   CALL KANAZAWA(MAIN) <+E+>
     00030 ?
     9 <+E+>
     00040 ?
     1.,2.1,3.,6.3,4.,8.5 <+E+>
     A =  -0.0429      B =   2.1286
              1      1.000      2.100      0.014
              2      3.000      6.300     -0.043
              3      4.000      8.500      0.029
     READY
```

図２０　ライブラリィを結合して実行形式プログラムを登録、実行する例

でデータセットKANAZAWA・LOAD(MAIN)に登録（＃２）する例である。更に＃４では、その実行形式プログラムをＣＡＬＬコマンドで実行している。実行形式で登録する場合は、＃２のようにロードライブラリィのデータセット名を指定する際にメンバ名の入力を忘れてはならない。忘れると〝TEMPNAME〟というメンバ名が自動的に付けられる。なお、＃３では入力対象とするデータセットタイプ〝OBJ〟を入力する。

## ４－４　ＦＯＲＴ７ＣＬＧコマンド

センタで開発されたFORT7CLGコマンドは、一つのコマンドで翻訳、結合編集、実行が可能である。また相対形式、実行形式プログラムの保存、実行もできる。すなわち、ＲＵＮコマンドの手軽さとＦＯＲＴ７７、ＬＩＮＫコマンドの持つロードモジュールの登録機能およびＬＯＡＤＧＯコマンド、ＣＡＬＬコマンドの持つ実行機能を兼ね備えており、一旦覚えてしまえば１つのコマンドで全てのプログラム処理が可能な便利なコマンドである。

ここではＦＯＲＴ７ＣＬＧコマンドの主な機能について解説する。詳細はセンタ発行の〝利用の手引き　ユーザコマンド編〟（車古　正樹・沼田　道代著）〔１３〕および〝センタコマンド編〟（車古　正樹著）〔１４〕を参照されたい。

（1） コマンドの説明

FORT7CLGコマンドの入力形式は以下のとおりである。

FORT7CLG データセット名1(メンバ名) 〔OP( {COMP RM LM LOAD CALL} )〕 〔LO(データセット名2) 〔M(メンバ名)〕〕〔L(区分データセット名)〕 〔IN(データセット名3(メンバ名))〕 〔OUT(データセット名4)〕

・データセット名1(メンバ名)はソースプログラムを翻訳、結合編集、実行する場合あるいはロードモジュールを登録する場合はそれが保存されている入力データセット名とメンバ名を指定する（データセットタイプFORT77あるいはFORT）。一方、ロードモジュールを実行する場合はそれが保存されているロードモジュールデータセット名とメンバ名を指定する（データセットタイプLOAD）。位置オペランドであり、省略できない。

・オプションオペランドOP( )はコマンドの持つ種々の機能を選択して指定する。括弧内の各オペランドの意味は

COMP…翻訳のみを行い、プログラムの文法エラーをチェックする。

RM …相対形式でロードモジュールを登録する。

LM …実行形式でロードモジュールを登録する。

LOAD…相対形式ロードモジュールを実行する。

CALL…実行形式ロードモジュールを実行する。

このオペランドを省略するとデータセット名1（メンバ名）で指定したプログラムを単に翻訳、結合編集、実行する。

・LO(データセット名2)オペランドは、相対形式又は実行形式で登録する場合に、そのロードモジュールを保存するデータセット名を指定する。既存のものでも新規に作成するものでも良い。メンバ名を入力してはならない。このオペランドの指定はOP（RM）又はOP（LM）を指定した場合のみ有効となる。

・M(メンバ名)オペランドは実行形式で登録したい場合あるいは相対形式で主プログラムを登録したい場合に、保存するメンバ名を指定する。相対形式でサブルーチンを登録する場合は、サブルーチン名が自動的にメンバ名として登録されるので、メンバ名を指定してはならない。

・L( )オペランドは結合時に必要なロードライブラリィのデータセット名を指定する。

・IN(データセット名3(メンバ名))オペランドは入力ファイル FT05F001を指定したデータセット名3（メンバ名）に割り当て、READ（5, …）に対する入力データをデータセットから読み込む場合に指定する（データセットタイプはDATA）。

・OUT(データセット名４)オペランドは出力ファイル　ＦＴ０６Ｆ００１を指定したデータセット名４に割り当て、ＷＲＩＴＥ（６，…）による出力結果をデータセットに書き込む場合に指定する（データセットタイプはOUTLIST）。

単にソース・プログラムを実行したい場合は

FORT7CLG　データセット名(メンバ名)　＜＋Ｅ＋＞

と入力すれば良い。また文法エラーのチエックのみを行ないたい場合は

FORT7CLG　データセット名(メンバ名)　OP(CO)　＜＋Ｅ＋＞

と入力すれば良い。なおＦＯＲＴ７ＣＬＧコマンドは、ＲＵＮサブコマンドで実行できないような大きなプログラムの実行も可能である。以下に実際の使用例を示す。

（２）　実行形式の登録と実行

ＦＯＲＴ７ＣＬＧコマンドで実行形式プログラムを保存し、かつ実行する例を図２１に示す。入力の対象としたのは図１７に示した最小二乗法による直線近似のプログラムである。ソースプログラムはKANAZAWA.FORT77(LINEFIT)に保存されており、作成した実行形式プログラムは新規の区分データセットISHIKAWA.LOADにメンバ名ＬＦＴとして登録するものとする。実行形式で登録する場合は図２１の＃１のように

FORT7CLG　データセット名１(メンバ名)　OP(LM)　LO(データセット名２)
M(メンバ名)　＜＋Ｅ＋＞

と入力すると、コマンドのスタート、ＦＯＲＴＲＡＮ７７コンパイラの呼び出し、翻訳の終了のメッセージが出力される。文法エラーがあると、それがここで表示され処理は中止される。翻訳が正常に終了すると結合が開始され、＃１のMオペランドで指定したメンバ名で実行形式プログラムがライブラリィに付加された由のメッセージが出力される。その後、さらに実行するかどうかを問い合せてくるので＃２のように"YES"と入力すると、すぐに実行に移され＃３のようにデータ入力を要求してくるので、ＲＵＮコマンドなどの場合と同様に対処すればよい。実行が終了すると＃４のメッセージが出力される。なお登録のみを行ない、実行したくない場合は"NO"と入力すればよい。

次に念のため＃５のようにＬＩＳＴＤＳコマンドでＩＳＨＩＫＡＷＡ．ＬＯＡＤのメンバ名一覧を出力すると、ロードモジュールの存在が確認できる。なお、＃６ではFORT7CLGコマンドによって実行形式プログラムISHIKAWA.LOAD(LFT)を再度実行している。その入力形式の一般形は次のとおりである。

FORT7CLG　データセット名(メンバ名)　OP(CALL)　＜＋Ｅ＋＞

```
     READY
#1   FORT7CLG KANAZAWA.FORT77(LINEFIT) OP(LM) LO(ISHIKAWA) M(LFT) <+E+>
     FORT7CLG STARTED TIME = 17:20:02
     FORTRAN 77 COMPILER ENTERED
     END OF COMPILATION (OPTIMIZATION COMPILER)
     LINKAGE START ***LIB('SYS1.SSL','APPF.LINKLIB')***
     **MEMBER NAME** LFT     WAS NOT FOUND BUT HAS BEEN ADDED TO LIBRARY.
     LINKAGE EDITER ENDED TIME=17:20:40
     ENTER YES THEN RUN.ELSE ENTER NO
#2   YES <+E+>
#3    00030 ?
     3 <+E+>
      00040 ?
     1.,2.1,3.,6.3,4.,8.5 <+E+>
      A =   -0.0429      B =    2.1286
                 1      1.000      2.100      0.014
                 2      3.000      6.300     -0.043
                 3      4.000      8.500      0.029
#4    ENDED CODE=0
      READY
#5   LISTD ISHIKAWA.LOAD M <+E+>
      AB9999.ISHIKAWA.LOAD
      --RECFM-LRECL-BLKSIZE-DSORG
        U      **     13030    PO
      --VOLUMES--
        USERD4
      --MEMBERS--
        LFT
      READY
#6   FORT7CLG ISHIKAWA(LFT) OP(CALL) <+E+>
      00030 ?
     3 <+E+>
      00040 ?
     3.,2.,4.,5.,6.,4. <+E+>
      A =    1.5000      B =    0.5000
                 1      3.000      2.000     -1.000
                 2      4.000      5.000      1.500
                 3      6.000      4.000     -0.500
      FORT77 GO ENDED TIME=17:22:01
      ENDED CODE=0
      READY
```

図２１　ＦＯＲＴ７ＣＬＧコマンドによる実行形式の登録と実行の例

（３）　相対形式の登録とライブラリィ指定による実行

次に図１７に示した最小二乗近似のソースプログラムの主プログラムとサブルーチンＬＳＱを相対形式でデータセットＩＳＨＩＫＡＷＡ．ＬＯＡＤに登録し、その主プログラムを実行する例を図２２に示す。相対形式で登録する場合の入力形式は

ＦＯＲＴ７ＣＬＧ　データセット名１（メンバ名）　ＯＰ（ＲＭ）　ＬＯ（データセット名２）
〔Ｍ（メンバ名）〕　＜＋Ｅ＋＞

```
     READY
#1   FORT7CLG KANAZAWA.FORT77(LINEFIT) OP(RM) LO(ISHIKAWA) <+E+>
     FORT7CLG STARTED TIME = 11:15:36
     FORTRAN 77 COMPILER ENTERED
     END OF COMPILATION (OPTIMIZATION COMPILER)
     LINKAGE START ***LIB(  'APPF.LINKLIB') ***
#2   JQA0461    JZLINIT*
#2   JQA0461    JZLLSTI*
#2   JQA0461    JZLSFMO*
#2   JQA0461    JZLSTOP*
#2   JQA0461    LSQ
#3   **MEMBER NAME** MAIN    WAS NOT FOUND BUT HAS BEEN ADDED TO LIBRARY.
#2                          DIAGNOSTIC MESSAGE DIRECTORY
#2      JQA0461-W WARNING : SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL-
#2                REFERENCE NCAL WAS SPECIFIED OR MARKED FOR RESTRICTED
#2                NO-CALL OR NEVERCALL.
#3   **MEMBER NAME** LSQ     WAS NOT FOUND BUT HAS BEEN ADDED TO LIBRARY.
#4   ENDED CODE=4
     READY
#5   FORT7CLG KANAZAWA.LOAD(MAIN) OP(LO) L(ISHIKAWA) <+E+>
     FORT7CLG STARTED TIME = 11:17:24
     00030 ?
    3 <+E+>
     00040 ?
    1.,2.1,3.,6.3,4.,8.5 <+E+>
     A =  -0.0429      B =   2.1286
              1      1.000      2.100      0.014
              2      3.000      6.300     -0.043
              3      4.000      8.500      0.029
     ENDEDE CODE=0
     READY
```

図２２　ＦＯＲＴ７ＣＬＧコマンドによる相対形式の登録とライブラリィ指定による実行の例

である。主プログラムを含むソースプログラムを登録する場合のみ、そのメンバ名をＭ(　)オペランドで指定する。主プログラムの登録の際にＭ(　)オペランドを省略するとＭＡＩＮというメンバ名で登録される。同一メンバ名が存在すると置換えられる。例ではまずＦＯＲＴ７ＣＬＧコマンドを図２２の＃１のように入力する。＃２、＃３のように結合時のエラーメッセージと登録完了のメッセージが出力され処理は終了する（＃４の完了コード４は正常終了を示す。なお実行はされない）。次に＃５のように再びＦＯＲＴ７ＣＬＧコマンドを入力し、相対形式プログラムを実行する。その入力形式は

ＦＯＲＴ７ＣＬＧ　データセット名(メンバ名)　ＯＰ(ＬＯＡＤ)　〔Ｌ(区分データセット名)〕
＜＋Ｅ＋＞

サブルーチンを結合する場合は、＃５のようにＬ(　)オペランドでライブラリィＩＳＨＩＫＡＷＡ．ＬＯＡＤを指定する。

（４）　大きいプログラムの実行

```
     READY
    E KANAZAWA(MICA) FORT77(FIXED) <+E+>
     E
    L <+E+>
#1   00100        DIMENSION A(100,100),B(100,100),C(100,100)
     00200        DIMENSION D(100,100),E(100,100)
     00300        DO 10 I=1,100
     00400         DO 20 J=1,100
     00500           A(I,J)=1.
     00600           B(I,J)=2.
     00700           C(I,J)=3.
     00800           D(I,J)=4.
     00900           E(I,J)=5.
     01000    20    CONTINUE
     01100    10  CONTINUE
     01200        WRITE(6,*)  A(1,1),B(100,1),C(1,100)
     01300        WRITE(6,*)  D(100,100),E(1,1)
     01400        STOP
#2   01500        END
     END OF DATA SET
     E
#3  RUN <+E+>
     FORTRAN 77 COMPILER ENTERED
     FORTRAN 77 ERROR MESSAGES: PROGRAM NAME(MAIN  ),FLAG(I),OPTIMIZE(2)
#4   JZK955I-U ID  008192            LACKED COMPILER WORKING AREA
     JZK904I-U             COMPILER ABEND
     RUN TERMINATED ABNORMALLY+
    END N <+E+>
     READY
#5  FORT7CLG KANAZAWA.FORT77(MICA) <+E+>
     FORT7CLG STARTED TIME = 09:00:03
     FORTRAN 77 COMPILER ENTERED
     END OF COMPILATION (OPTIMIZATION COMPILER)
     LINKAGE START ***LIB('SYS1.SSL','APPF.LINKLIB')***
     **MEMBER NAME**MAIN   WAS NOT FOUND BUT HAS BEEN ADDED TO LIBRARY.
#6   1.00000000 2.00000000 3.00000000
     4.00000000 5.00000000
     ENDED CODE=0
     READY
```

図２３　ＦＯＲＴ７ＣＬＧコマンドによる大きいプログラムの実行例

図２３にＲＵＮサブコマンドで実行できない大きなプログラムをＦＯＲＴ７ＣＬＧコマンドで実行する例を示す。ソースプログラムは既存データセットKANAZAWA.FORT77(MICA)にある（＃１～＃２）。ＥＤＩＴのサブコマンドで＃３のようにＲＵＮサブコマンドを入力して実行すると、＃４の表示のように作業領域不足のため翻訳の段階で異状終了する。そこでエディットモードを終了し、＃５のようにＦＯＲＴ７ＣＬＧコマンドで実行してみると、＃６のように出力結果が得られ、実行が可能である。

## 4－5　バッチジョブの依頼

本章の前節までに述べたコマンドによる実行とは全く別の方法として、ＳＵＢＭＩＴコマンド（又はＳＵＢＭＩＴサブコマンド）によって会話型リモートバッチジョブを依頼する方法があることは利用者の周知のとおりである。ＣＰＵタイムを要するジョブはバッチジョブを依頼する方が料金的に有利である。しかし、前節までに述べた種々の実行方法を修得すると、利用の手引き〔2〕で修得したように１つのデータセットにＪＣＬ文、ソースプログラム、実行データを入れておくことはいろいろと不都合を生じてくる（たとえば、直接ＲＵＮサブコマンドなどで実行できない）。そこでＪＣＬ文のみをデータセットタイプ″CNTL″のデータセットに準備し、入出力ファイルを定義するＤＤ文（ＪＣＬ文の一種）でデータセットタイプ″FORT77(FIXED)″、″DATA″なるデータセットに保存されているソースプログラムや実行時の入力データを結合する方法が有用となってくる。１つのデータセットにＪＣＬ文、ソースプログラム、実行時の入力データが既に入っている場合は、データセットタイプが″ＦＯＲＴ７７（ＦＩＸＥＤ）″や″ＤＡＴＡ″の新しいデータセットをEDITコマンドで開き、MERGEサブコマンドを用いて必要な部分を移せばよい（ＭＥＲＧＥサブコマンドについては３－４（４）参照）。

実行時の入力データやソースプログラムが別々のデータセットに存在する場合のＪＣＬ文の一般的並びを図２４に示す。これをデータセットタイプ″CNTL″に作成し、サブミットすれば良い。なお各行の先頭に付けた<1>～<4>の番号は説明のために付したものであり入力してはならない。

```
<1> //AB9999X JOB ,PASS=パスワード,CLASS=A,MSGCLASS=D
<2> // EXEC FORT7CLG〔,L='AB9999.ユーザ指定名１.LOAD'〕
<3> //FORT.SYSIN DD DSN=AB9999.ユーザ指定名２.FORT77(メンバ名),
<3> //     DISP=SHR
<4> //GO.FT05F001 DD DSN=AB9999.ユーザ指定名３.DATA(メンバ名),
<4> //     DISP=SHR
    //
```

図２４　サブミットジョブのＪＣＬ文

図２４解説

<1>：ジョブ文。ＡＢ９９９９Ｘは課題番号プラス任意の一文字を指定する。ＣＬＡＳＳはＡ～Ｄ中から適当なものを選択する。

<2>：ＥＸＥＣ文。Ｌ＝…以下は相対形式で登録されたライブラリィの結合が必要な場合のみ、そのデータセットの完全修飾名を引用符でくくって指定する。大括弧〔　〕はコマンドの表示同様省略可能であることを表す。省略の際は″Ｌ＝″の前のコンマ″，″もとることを忘れないように！

<3>：ソースプログラムの入力先を定義するＤＤ文。ソースプログラムが保存されているデータセットの完全修飾名とメンバ名を指定する。引用符でくくってはならない。

<4>：実行データの入力先を定義するＤＤ文。データの保存されているデータセットの完全修飾名とメンバ名を指定する。入力データがない場合はこのＤＤ文を省略して良い。

さて実行時の入力データをＪＣＬ文に含めたい場合、従来と同様　図２４の<4>のかわりに

```
//GO.SYSIN DD *
 入力データ
/*
```

とすれば良い。一方、ソースプログラムあるいは実行時の入力データが複数のデータセットに存在する場合は、<3>のかわりに

```
//FORT.SYSIN DD DSN=データセット名２(メンバ名),DISP=SHR
//           DD DSN=データセット名４(メンバ名),DISP=SHR
          :
```

のように、又<4>のかわりに入力順に

```
//GO.FT05F001 DD DSN=データセット名３(メンバ名),DISP=SHR
//            DD DSN=データセット名５(メンバ名),DISP=SHR
          :
```

などのように、データセットを連結して指定すれば良い。

また、入出力の論理機番として５～７以外を使用する場合（たとえば、ＷＲＩＴＥ（１０，…）…）は、それに対応したＤＤ文<5>あるいは<6>を次の例のように<4>の後に定義すればよい。なお、<5>は順データセットを新規に確保する場合の例であり、<6>は既存のデータセットを指定する場合である。

```
<4>//GO.FT05F001 DD DSN=データセット名３,(メンバ名)、DISP=SHR
<5>//GO.FT10F001 DD DSN=データセット名６,UNIT=SYSDA,
<5>//    VOL=SER=USER01,DISP=(NEW,CATLG),SPACE=(TRK,
<5>// (10,10),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
<6>//GO.FT20F001 DD DSN=データセット名７,DISP=SHR
```

なお、カタログドプロシジャFORT7CLGのＪＣＬ文に関する詳細は〝利用の手引き　ＦＯＲＴＲＡＮ　カタログド・プロシジャ編〟（車古　正樹著）〔１５〕を参照されたい。

## おわりに

本稿は計算機センタ発行の二編の手引き〝ＴＳＳ実習編（初心者用）－ＦＯＲＴＲＡＮユーザのためのＴＳＳ入門－〟（中島　恵美著）〔１〕、〝ＴＳＳ端末によるバッチ処理－フルスクリーン機能と出力検索－〟（関崎　正夫著）〔２〕の後を受けて、ＴＳＳの初級ユーザが修得すべき基本的知識を中心にまとめたものである。記述にあたり、理解の容易さに重点をおく一方、そのために表現があいまいにならないようにも努めたつもりである。そのあまり書き終えてみると、利用の手引きとしては不適当な長さとなり後悔している。もし第４章まで順に読破されるユーザがあるならその忍耐と熱意に敬意を表したい。

さて、本手引きの後半で紹介した各種コマンドの機能は、それらのコマンドの持つ機能の主要部を抜き出したものであり、決してその全てを網羅したものではないことを、ユーザは強く認識して頂きたい。したがって、上に掲げた二編の手引きと本手引きの内容を修得したユーザは、富士通発行の〝ＴＳＳコマンド文法書〟〔４〕を始めとする各種マニュアルを時折開いてよりいっそうの知識の補充に努められることを勧める。また、ＴＳＳにはさらに便利な各種ソフト的機能が用意されており、それらに関して後続の利用の手引き〝ＴＳＳ実習編（中級用）〟〔８〕、〔１０〕を参照されたい。

## 関連マニュアル一覧


〔1〕 中島　恵美著、利用の手引き　ＴＳＳ実習編（初心者用）―ＦＯＲＴＲＡＮユーザのためのＴＳＳ入門―、金沢大学計算機センタ

〔2〕 関崎　正夫著、利用の手引き　ＴＳＳ端末によるバッチ処理―フルスクリーン機能と出力検索―、金沢大学計算機センタ

〔3〕 山崎　光悦著、利用の手引き　バッチからＴＳＳへ、金沢大学計算機センタ

〔4〕 ＦＡＣＯＭ　ＯＳＩＶ／Ｆ４　ＴＳＳコマンド文法書　Ｅ４０系用、富士通

〔5〕 ＦＡＣＯＭ　ＯＳＩＶ　ＴＳＳテキスト編集使用手引書　Ｆｕｌｌ　Ｓｃｒｅｅｎ　Ｏｐｔｉｏｎ（ＦＳＯ編）、富士通

〔6〕 ＦＡＣＯＭ　ＯＳＩＶ／Ｆ４　ＴＳＳ　ＳＯＲＰコマンド使用手引書、富士通

〔7〕 ＦＡＣＯＭ　ＯＳＩＶ／Ｆ４　ＴＳＳデータセットプリント（ＤＳＰＲＩＮＴ）使用手引書、富士通

〔8〕 関崎　正夫著、利用の手引き　ＴＳＳ実習編（中級用）　ＦＯＲＴＲＡＮ７７　インタラクティブデバッグ―実行時のエラーの原因を見つける―、金沢大学計算機センタ

〔9〕 ＦＡＣＯＭ　ＯＳＩＶ／Ｆ４　ＦＯＲＴＲＡＮ７７　インタラクティブディバッグ使用手引書、富士通

〔10〕 山下　邦弘著、利用の手引き　ＴＳＳ実習編（中級用）―ＦＯＲＴＲＡＮユーザのためのジョブ制御文とデータセット―、金沢大学計算機センタ

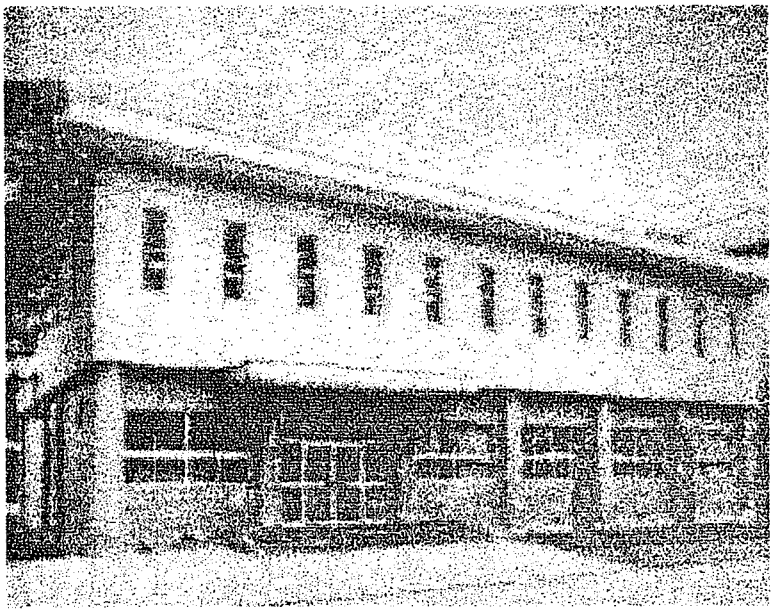
〔11〕 ＦＡＣＯＭ　ＯＳＩＶ／Ｆ４　データセット管理機能説明書、富士通

〔12〕 ＦＡＣＯＭ　ＯＳＩＶ／Ｆ４　ＦＯＲＴＲＡＮ７７　使用手引書、富士通

〔13〕 車古　正樹・沼田　道代著、利用の手引き　ユーザコマンド編、金沢大学計算機センタ

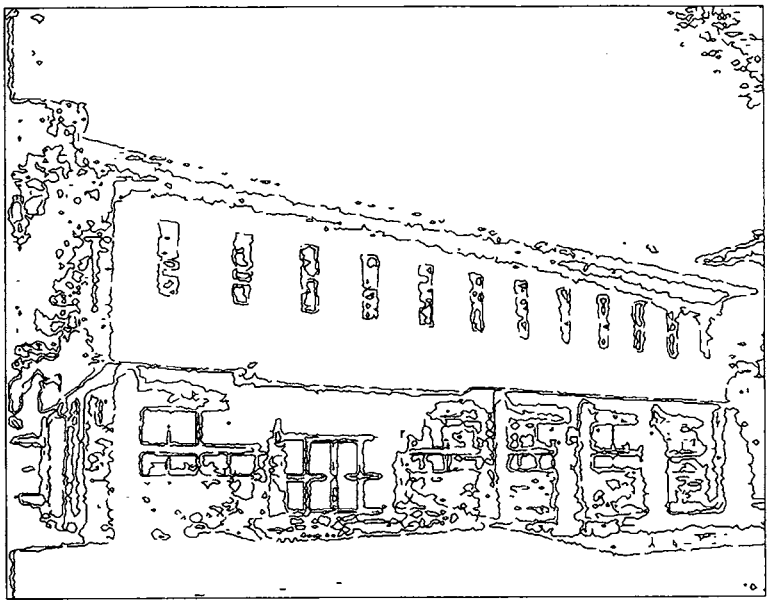
〔14〕 車古　正樹著、利用の手引き　ＴＳＳ実習編（全般用）―ＴＳＳユーザのための便利なコマンド―、金沢大学計算機センタ

〔15〕 車古　正樹著、利用の手引き　ＦＯＲＴＲＡＮカタログド・プロシジャ編、金沢大学計算機センタ



金沢大学計算機センター全景

この写真はＴＶカメラ図形入力システムによってディジタイズされた図形データを日本語ラインプリンタで出力したものです。



この図は上の写真をＳＡＳグラフ機能で等濃度処理したものです。